COMBAT IDENTIFICATION MODELING
USING NEURAL NETWORKS TECHNIQUES

THESIS

Changwook Lim, Captain, ROKA

AFIT/GOR/ENS/09-09

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/09-09

COMBAT IDENTIFICATION MODELING USING NEURAL NETWORKS TECHNIQUES

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Changwook Lim, BS

Captain, ROKA

March 2009

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GOR/ENS/09-09

COMBAT IDENTIFICATION MODELING USING NEURAL NETWORKS TECHNIQUES

Changwook Lim, BS
Captain, ROKA

Approved:

__________(Signed)_____________ __________ Mar-05-2009
Dr. Kenneth W. Bauer (Chairman) date

__________(Signed)______________________________ Mar-05-2009
Dr. J. O. Miller (Member) date

AFIT/GOR/ENS/09-09

Abstract

The purposes of this research were: (1) validating Kim's (2007) simulation method by applying analytic methods and (2) comparing the two different Robust Parameter Design methods with three measures of performance (label accuracy for enemy, friendly, and clutter). Considering the features of CID, input variables were defined as two controllable (threshold combination of detector and classifier) and three uncontrollable (map size, number of enemies and friendly).

The first set of experiments considers Kim's method using analytical methods. In order to create response variables, Kim's method uses Monte Carlo simulation. The output results showed no difference between simulation and the analytic method.

The second set of experiments compared the measures of performance between a standard RPD used by Kim and a new method using Artificial Neural Networks (ANNs). To find optimal combinations of detection and classification thresholds, Kim's model uses regression with a combined array design, whereas the ANNs method uses ANN with a crossed array design. In the case of label accuracy for enemy, Kim's solution showed the higher expected value, however it also showed a higher variance. Additionally, the model's residuals were higher for Kim's model.

AFIT/GOR/ENS/09-9

To My son and Wife

Acknowledgments

There have been a number of people without whom this thesis could not have been completed. My deepest gratitude goes out to Dr. Kenneth W. Bauer. Dr. Bauer has tolerated an incalculable number of questions from me and has weathered that storm beautifully. He has helped me to develop a deep interest in the research process and has mentored me well. I also would like to thank my committee member: Dr. J.O. Miller for his flexibility with me, his perceptive comments, and his ongoing effort to help me be a better writer.

I would like to express my undying gratitude to my wife, Hyejung Lee. Also, I wish to thank my classmates, Maj. Miller and Maj. Shilland.

Changwook Lim

Table of Contents

Page

List of Figures

Page

List of Tables

Page

COMBAT IDENTIFICATION MODELING USING
NEURAL NETWORK TECHNIQUES

I. Introduction

Background

"Historically, friendly fire incidents have accounted for about 15 percent of all casualties on the battlefield. Operation Desert Storm in 1991 was no exception and fratricide rates showed no improvement during the 2001 Division Capstone Exercise, a test of Army digitization. The Future Force will be equally vulnerable unless a reliable combat identification system is fielded. Friendly fire, or fratricide, incidents killed or injured about 17 percent of the American casualties during Operation Desert Storm in 1991" [14]. After the war in the Gulf, U.S. officials vowed to reduce the number of friendly fire incidents in future conflicts. The "100 hour" Desert Storm ground campaign explained the brutality and the high tempo of modern war. For several days, almost one million coalition forces and more than ten thousand armored vehicles engaged in an intense and continuous battle, often in rainy weather [14]. "Unlike previous conflicts where the front lines remained relatively fixed, Operation Desert Storm was characterized by a dynamic, often confused battlefield where individual combat vehicle crews and units, caught up in the rapid advance punctuated by pitched skirmishes and battles, sometimes lost their "situational awareness" of where they were and where the enemy and friendly forces were." [14]

Successful Combat Identification (CID) is a very important factor to success in various missions of combat. For instance, a reliable detection and classification of an enemy target is essential at the real battle field. Since modern enemies, such as al'Qaeda, tend to hide in cluttered urban areas, it is extremely hard to destroy them without civilian casualties and collateral damage. Thus, we need rapid, effective CID processing in order to succeed in future combat. A good method to assess the iterative CID process is simulation, and constructing appropriate prediction model of detection and classification is important, since wrong model could lead to fratricide in the complex battlefield.

Research Problem

In the fall of 1994, a DoD Combat CID Study was performed at the request of Dr. Paul Kaminski to do a DoD-wide review of CID, and this study was completed by the summer of 1995 [2]. The Defense Science Board Task Force concluded that there was no crisis in CID calling for extraordinary action and suggested the maintaining of current CID budgets and activities [2:45-47]. After the Task Force's report, CID has been investigated considerably, especially with respect to automatic target recognition (ATR). The study of the ATR model has been conducted by Dr. Bauer and his students at AFIT. And Dr. Bauer and Capt. Kim constructed a full process model of CID including ATR; however, the regression method used in the Kim's model is only linear. Artificial Neural Networks (ANNs) afford a richer representation and, as such, are the focus of this research.

Research Objective

In this paper we first need to validate Kim's simulation method. The method uses Monte Carlo simulation to create response variables. This research compares Kim's response variables to theoretical values, based on Bayes' theorem.

Also, this paper considers three measures of performance (label accuracy for enemy targets, label accuracy for friendly objects and label accuracy for clutter objects) in comparing Kim's regression and this new Artificial Neural Network (ANNs) method. Optimal points are determined by each method and contrasted through confirmation experiments.

Scope

This paper will mainly deal with validating Kim's simulation with probability theory and constructing a prediction model of CID and its evaluation techniques. In order to construct a prediction model, this research use only ANNs method, however, this research will motivate further research using different techniques.

Overview

The next four chapters provide detailed information and descriptions of this research. Chapter two summarizes the literature relating directly to this research. Chapter three explains the CID model established for this research and outlines the methodology used to perform the problem discussed in Chapter 1 and Chapter 2. Chapter four presents the description of experiments and the results of the analysis. Chapter five provides the author's conclusions and recommendations for future research.

II. Literature Review

Overview of Department of Defense Modeling and Simulation Pyramid

Modeling and Simulation (M&S) is defined as "The process of designing a model of a system and conducting experiments with this model for the purpose either of understanding the behavior of the system or of evaluating various strategies for the operation of the system" [1]. There are numerous reasons why computer simulation is used for modeling a system. For instance, simulation model could be quiet complex, if we need to represent a system in detail, however we can still analyze the complex model. And if a specific system requires dangerous or expensive situations in real world, then we should use computer simulation. Especially, it would be impossible and immoral to process a real combat in order to simply test a new weapon [4:5].

A model of a real system is a representation of some of the components of the system and of some of their actions and interrelationships which are useful for description or forecast the behavior of the system [6: Sec I, 1].

Model Hierarchy

Combat models use a multi-tiered hierarchical family of models [3]. The bottom of the pyramid is a high resolution combat model including the detailed interactions of individual combatants or weapons. The focus on details makes high resolution models as reasonably credible representation of combat, but also limits high-resolution models to fairly small forces [6: Sec I,3].

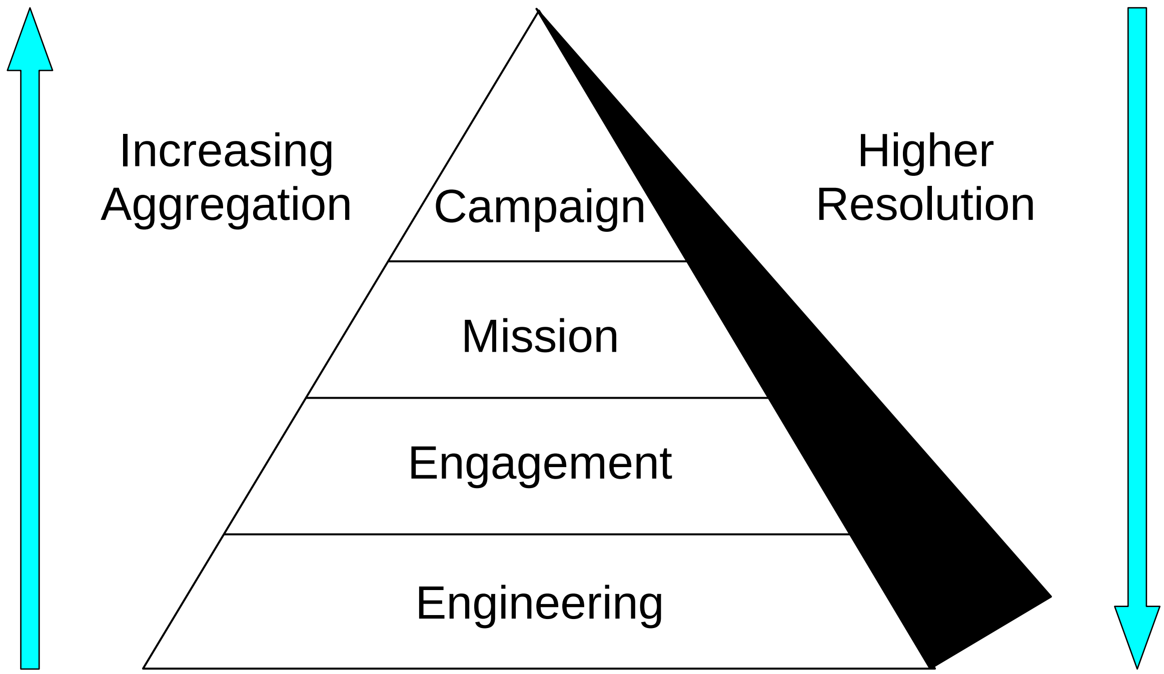


Figure 1: DoD M&S Pyramid [3]

Since the primary model applied in this research considers the engagement and battles, a high resolution model is designed to determine the operational performance of the system.

Description of CID Mission

Definition

CID is the process of achieving an accurate characterization of entities in a combatant's area the responsibility to the extent that high confidence, real-time application of tactical options and weapon resources can occur. The objective of CID is to maximize control and mission effectiveness, while reducing the total number of victims as a result of enemy action and fratricide [2:1].

Importance of effective CID

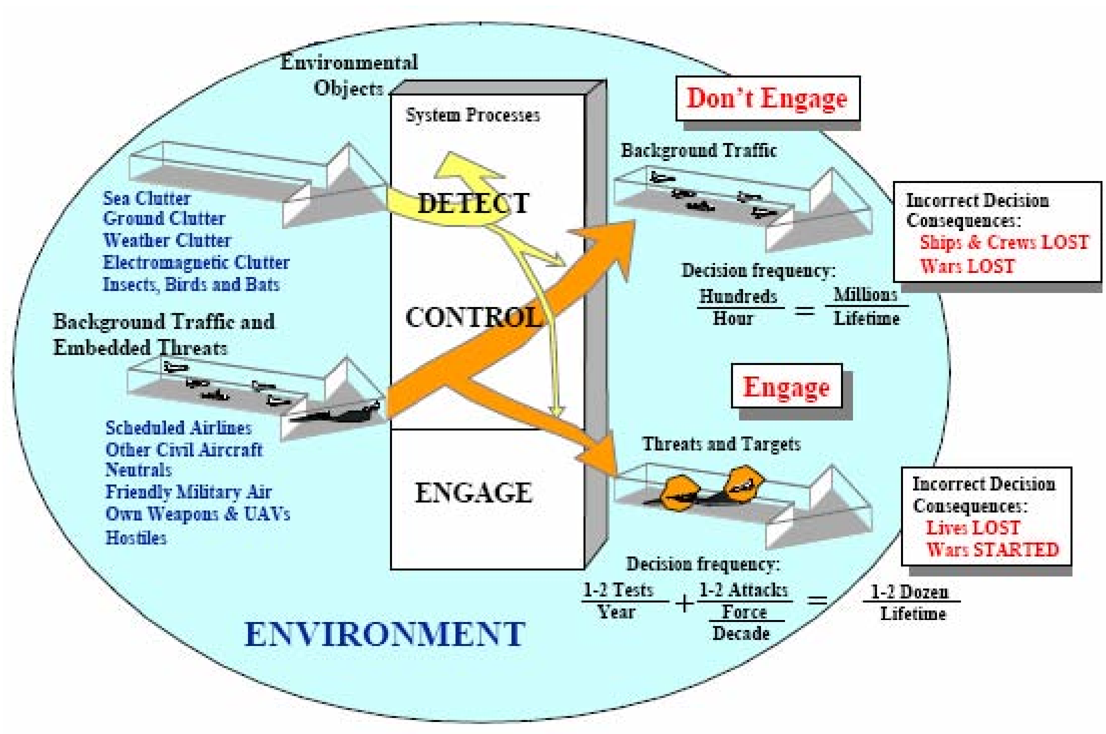


Figure 2: Importance of Effective CID [5:4]

Figure 2 shows why execution of a correct CID is important. If an object is enemy, but not identified as hostile, and thus the Blue force does not destroy it, ships and crews of the Blue force may be lost, eventually wars would be lost. Furthermore, if the object is friendly or civilian and the Blue force destroys the result of a false identification, then lives are lost and wars can be started [5:3]

Areas of CID Scenarios

CID for real time target identification to combatants has four mission areas: (1) surface-to-surface, (2) air-to-surface, (3) surface-to-air, and (4) air-to-air. Figure 3 shows the difference of proportions between the old wars and a recent war. Those percentages are changed a only slightly, while many years are passed. 'Operation Desert Storm' indicates the importance of the surface-to-surface CID missions, however, it is hard to say that the surface-to-surface mission is the most essential part of CID, since the importance of CID mission can be changed in the environment of battlefield. For instance, air-to-surface can be the most important mission area of CID where targeting on ground is impossible or an aircraft fires directly after targeting, involving the collateral damage.

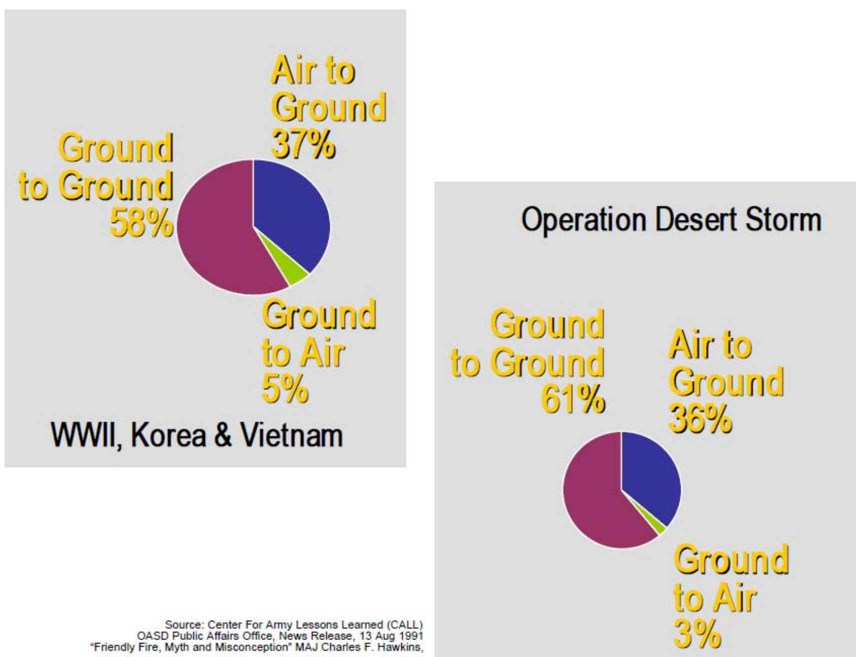


Figure 3: The proportions of CID mission [16]

Constructing a model

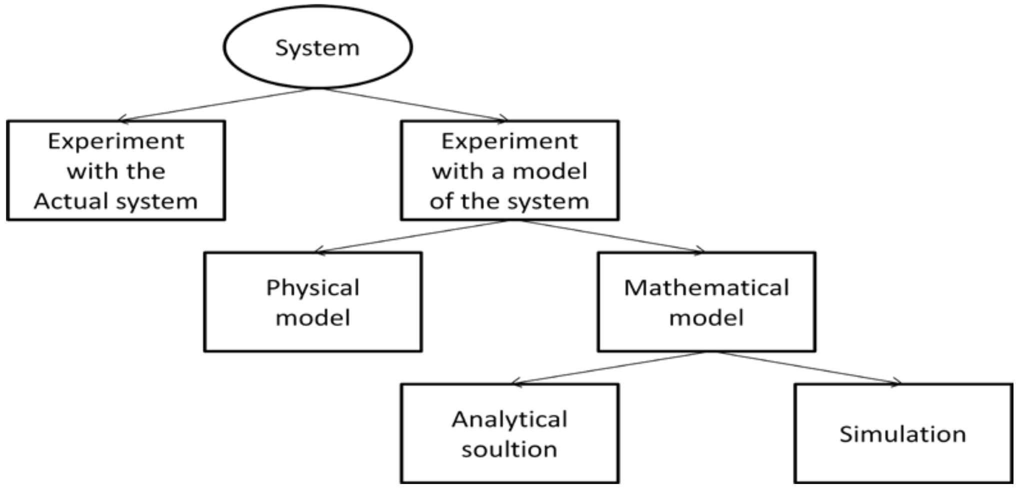


Figure 4 : Ways to study a system[8]

If an experiment with model of a system is possible, we can build a mathematical model, it must then be checked to see how it can be used to answer the questions of interest about the system it is supposed to represent. If the model can be represented in a simple form, it may be possible to get an *analytic* solution. When an analytical solution of a mathematical model is available and is computationally efficient, it is usually desirable to model the system in this way rather than through a simulation [8].

Analytic Model

An analytical model consists of an explicit mathematical formula for each of the output variables as a function of the only input variables. Analytical solutions are obtained by using the rules of mathematics to manipulate the equation of the model with the achievement of the required output formats. Analytical solutions are desirable, since the relationship between input and output is shown as an explicit and hopefully simple formula. An analytic solution will typically consist of; (1) an explicit formula for the probability of the output variable, or (2) an explicit formula for the mean value of the output variable [6: Sec I, 6].

Simulation Model

A simulation model solution is obtained by sequential action of the processes and interactions of the model. This is usually done with a digital computer, so that simulation models are particularly suitable for the models whose relationships are expressed in a procedural rather than algebraically. Simulation is the solution method that can best deal with complex, dynamic, high resolution models of force-on-force combat where simplifying assumptions would seriously disrupt the model of the representation of the real world system [6: Sec I, 7].

A common problem in many defense decision-making contexts that "modeling" is combined with "simulation." Although an increasing number of operational and executive decisions depend on the results of a growing list of large, complex computerized renditions of combat, a small number of the analysts who use these "simulations" fully understand the mathematical relations, or models, that drive them. This may lead to a false sense of formality and the validity to the decisions the models support. Analysts often approve the analysis results "from the simulation," as if that fact alone has analytical validity. The match between the mathematical guts of a simulation and the structure of the problem being simulated is often ignored. Despite the importance of verification, validation and accreditation (VV & A), simulation VV & A is inconsistently applied in practice - especially with regard to the suitability of mathematical models to real-world processes [17:2].

Validating the Output from the Overall Simulation Model

The most definitive test of the validity on a simulation model is to establish that its output data closely resembles the output data expected from the actual system. This is called ‘results validation’, and there are several ways this can be implemented [8:259].

Comparison with an Existing System

If a system under study is similar to an existing system, then a simulation model of the existing system can be developed and its output data compared to those from the existing system itself. If the output data from two sets are closely matched, the model of the existing system is valid. The comparison of the model and system output data could be done using the numerical statistics such as the mean, variance and correlation function. Alternatively, the assessment can be made using graphs such as histograms, distribution functions, and plots with ‘Microsoft Excel’ or ‘MATLAB’ [8:259].

Comparison with Expert Opinion

Regardless of existence of a system, experts of simulation should review the simulation results for reasonableness. If the simulation results are consistent with perceived system behavior, then the model can be said to have ‘face validity’.

Comparison with Another Model

If another model was developed for the same system and for a similar purpose, then it could be a valid representation. Numerical statistics or graphical plots with ‘Microsoft Excel’ or ‘MATLAB’ can be a method for comparing two models. However, even if the two models produce similar results, we cannot say the model is necessarily valid, since both models could have a similar error [8:263]. An analytic model is used in this research in order to validate Kim’s simulation with Baye’s rule, and described later in the methodology. This research constructs a new model of the system using the ANN method, then compares results between Kim's model and the ANN model. The methods and experiments will be explained in later chapters.

Animation

An animation can be an effective way to find invalid model assumptions and improve the credibility of a simulation model [8: 264].

Receiver Operating Characteristics Curve

Receiver Operating Characteristics (ROC) analysis are used to describe the tradeoff between true positive rate (TPR) and false positive rate (FPR) in signal detection theory. Besides being a commonly useful performance measure, ROC analyses are especially useful when observing skewed class distribution and different classification error costs. These properties are very important in the area of cost-sensitive learning and learning in the presence of unbalanced classes [7:1].

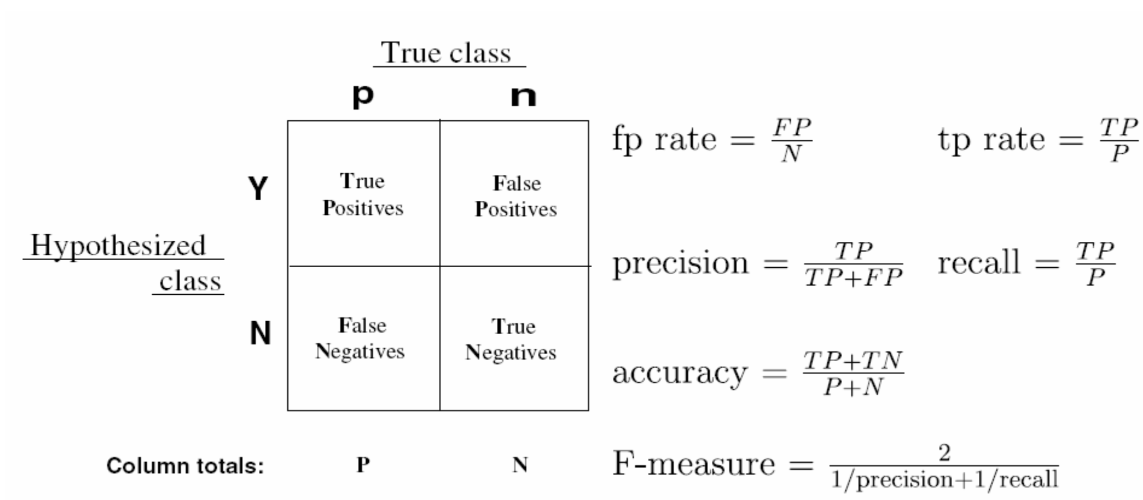


Figure 5: Confusion Matrix and Common Performance Metrics [7:2]

Figure 5 shows four possible outcomes based on classifier and instance. ‘Y’ and ‘N’ mean the hypothesized declaration of positive or negative, relative to some target class, that is, ‘Y’ is positive output of simulation. ‘N’ is negative output of simulation. ‘p’ and ‘n’ denote the true class. If the true class is positive and its simulation output is also positive, it is a true positive; if the predicted output is negative it is a false negative. If the true class is negative and the simulation output classified as negative, it is a true negative; if the predicted output classified as positive, it is a false positive. A set of true classes and predicted classes can be used to construct a two-by-two confusion matrix (CM).

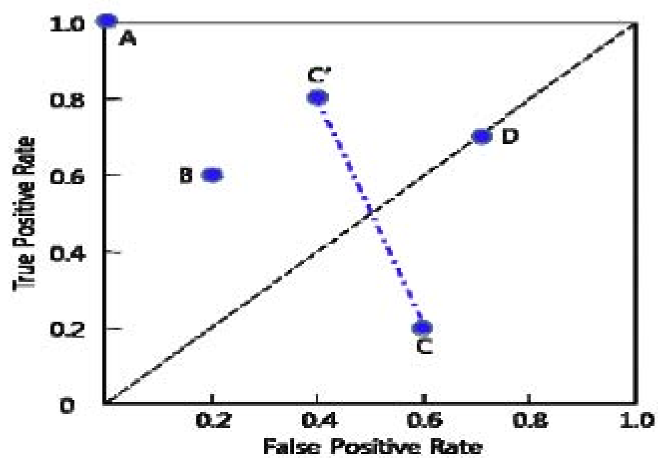


Figure 6: ROC Space Graph

ROC graphs have two dimensions in which the Y axis is true positive (TP) rate and the X axis is false positive (FP) rate. Figure 6 shows ROC space with five discrete classifiers generating a (FP rate, TP rate) pair corresponding to its class value. Point A, (0, 1) represents perfect positive classification. This point is the best possible prediction, representing 100% sensitivity (recall) and specificity (1-*fp* rate). Performance in the northwest (FP low, TP high), represents the best classification. Classifiers appearing on the left-hand side of the ROC graph, near the X axis, may be thought of as "conservative": they make positive classifications only with strong evidence so they make few false positive errors, but they often have low TPRs as well [7:3]. Classifiers on the upper right-hand side of an ROC graph may be thought of as "liberal": they make positive classifications with weak evidence so they classify nearly all positives correctly, but they often have high FPRs [7:3]. In Figure 6, B is more conservative than C′.

The point D on the diagonal line represents completely random guess. And the point C located in the lower right triangle shows worse performance than random guess. The relation between point C and C′ shows an opposite condition of classification output on every true class – its TP rate becomes false negative rate (FNR) and its FP rate becomes true negative rate (TNR). Hence, point C in the lower right triangle is negated to point C′ in the upper left triangle.

What methods are used for Combat Identification

Monte Carlo Simulation and Regression (Kim 2007))

A Monte Carlo simulation can be defined as a model using random numbers, that is, U(0, 1) random variates. It is used for solving stochastic or deterministic problems [5:73]. The name "Monte Carlo" simulation is derived from World War II, and Monte Carlo simulation is widely applied for solving statistics problems that are not analytically tractable [8:74]. Since Monte Carlo simulation has repeated calculations of random numbers, it is suitable in a computer calculations as Kim made MATLAB code in his thesis [4:23].

In order to make a prediction model Kim focused on the linear regression models. The general regression model is represented by equation (2.1).

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon \qquad (2.1)$$

Where y is the response variable, $\beta_j s(j = 0,1,...,k)$ are regression coefficients and $x_i s(i = 0,1,...,k)$ are predictor variables [11:374]. This research will explain Kim's method in following chapters.

Bayesian Networks

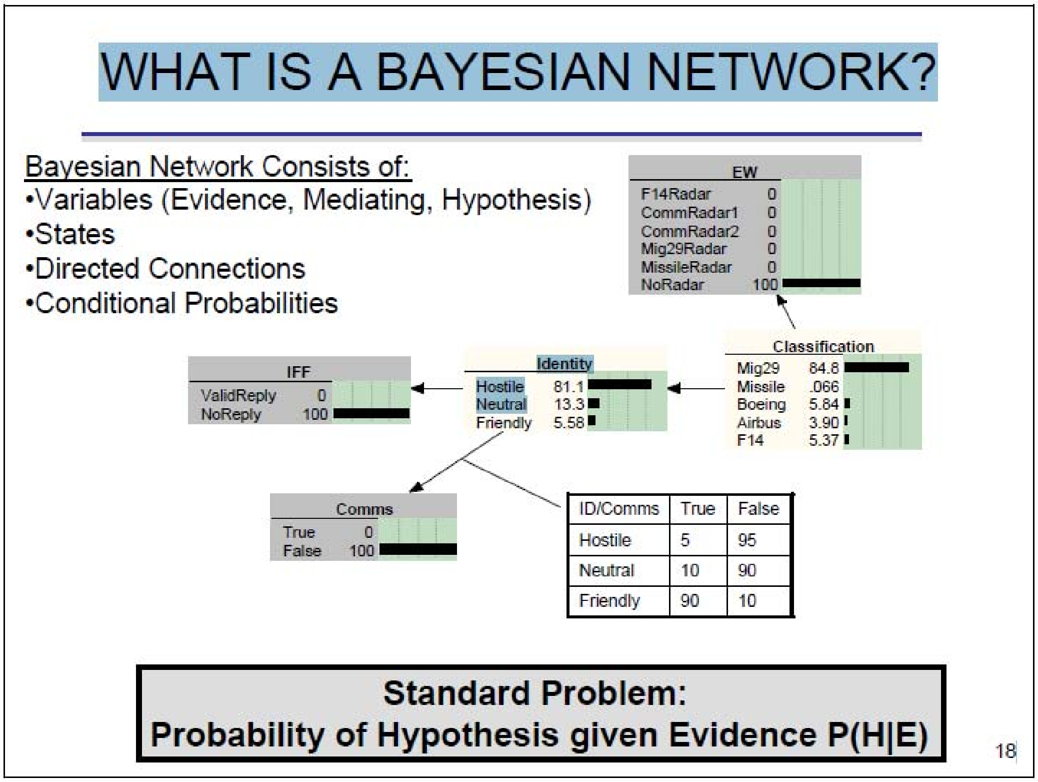


Figure 7: Example of Bayesian Networks [5:7]

Figure 7 is an example of a Bayesian network. The standard problem involving a Bayesian network is the calculation of the probability of the hypothesis of different states through various mediating variables. Bayesian networks are easy to create or modify. Bayesian networks can mix historical modeling and simulation, and expert judgment. The structure and parameters can be drawn from data. They offer several advantages over standard statistical techniques since they use conditional independence to reduce the number of estimation parameters. Since efficient algorithms were developed in the late 1980s for the calculation of probability, they are easy to operate. These graphical models are more understandable than neural networks [5:6].

Mathematical Frame Work for CID Simulation [4:27-28]

Kim constructed confusion matrices (CM) of the detection, classification and overall CID system, since both detection and classification are essential parts of CID.

Table 1: Detection, Classification and System Confusion Matrices

	True Classes		
Detector "Labels"	Enemy or Friend	Clutter	Horizontal Totals
"Enemy or Friend"	E or F labeled "EF"	C labeled "EF"	"E or F" Declared
"Clutter"	E or F labeled "C"	C labeled "C"	"C" Declared
Vertical value	E or F evaluated	Clutter evaluated	

		True Classes		
Classifier "Labels"	Enemy	Friend	Clutter	Horizontal Totals
"Enemy"	E labeled "E"	F labeled "E"	C labeled "E"	"E" Declared
"Friend"	E labeled "F"	F labeled "F"	C labeled "F"	"F" Declared
Vertical value	Enemy evaluated	Friend evaluated	Clutter evaluated	

		True Classes		
System "Labels"	Enemy	Friend	Clutter	Horizontal Totals
"Enemy"	E labeled "E"	F labeled "E"	C labeled "E"	"E" Declared
"Friend"	E labeled "F"	F labeled "F"	C labeled "F"	"F" Declared
"Clutter"	E labeled "C"	F labeled "C"	C labeled "C"	"C" Declared
Vertical value	Enemy evaluated	Friend evaluated	Clutter evaluated	

The above three tables show a CM of the detection process (top), that of classification process (middle) and that of the system (bottom). The color of each cell between the three tables shows the relationships between the cells, since the classification depends on the output results of the detection process, which is to say that something must be detected before proceeding in the classification process. We see all the detected simulation output of the detection on the classification CM. The sum of the same colors on the system will coincide with the graph shown on the detection process CM table of the same color. Kim calculated TPR, E_{CR} (critical error) and label accuracy.

The TPR for a enemy is $P("E"|E)$, the probability of labeling enemy given true enemy. The equation for this probability is

$$P("E"|E) = \frac{P("E" \cap E)}{P(E)} \approx \frac{\text{first row and column of system's CM}}{\text{sum of first column of system's CM}}. \quad (2.2)$$

The E_{CR} (FPR), the probability true friend given labeled enemy for fratricide, can be represented in the similar manner.

$$P(F|"E") = \frac{P(F \cap "E")}{P(E)} \approx \frac{\text{first row and second column of system's CM}}{\text{sum of first row of system's CM}} . \quad (2.3)$$

The E_{CR} is represented in horizontal analyses of the CM frequency counts. In this effort, Kim also defined the label accuracy which is actually needed by a warfighter before he makes fire decision.

$$P(E|"E") = \frac{P(E \cap "E")}{P("E")} \approx \frac{\text{first row and column of system's CM}}{\text{sum of first row of system's CM}} . \quad (2.4)$$

The Neural Network

"Neural Nets can be classified in a systematic way as systems or models composed of "nodes" and "arcs", where the nodes are artificial neurons or *units* (in order to distinguish them from their biological counterparts, which they mimic only with respect to the most basic features). Usually, within a specific NN all units are the same. The arcs, or connections between the units, simultaneously mimic the biological *axons* and the *dendrites* (in biology, the fan-in or input-gathering devices) including the *synapses* (i.e. the information interface between the firing axon and the information-taking dendrite). Their artificial counterpart is just a "weight" (given by a realvalued number) that reflects the strength of a given "synaptic" connection" [9:8]. The type of connection is the basis for the enormous diversity in NN architectures, with great diversity in their behavior. Figure 8 shows the described relationships between the biological neuron and its artificial counterpart, the unit [9].

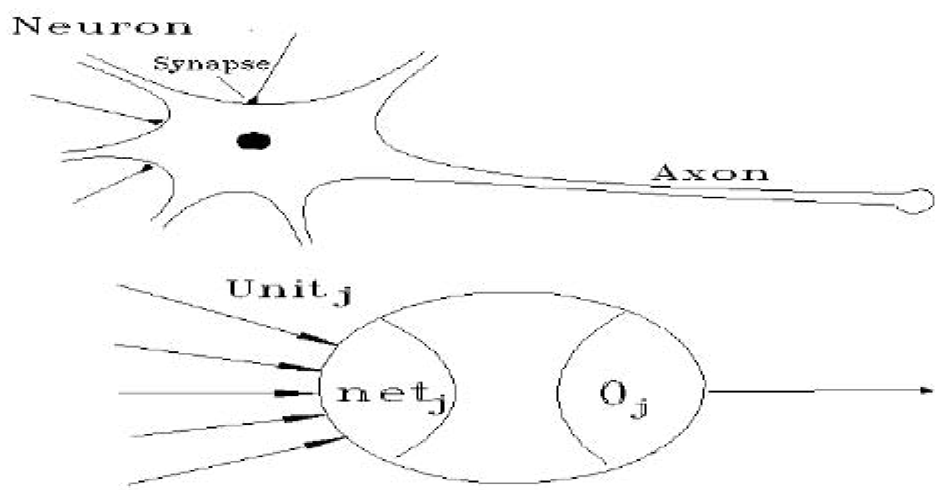


Figure 8: Neuron and Unit [9]

Artificial neural networks are an active area of research and application, in particular for the analysis of large, complex, highly nonlinear problems [13: Sec9.7].

The advantages of neural networks are follows [15]:

- The principal advantage of neural networks is that it is possible to train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. For example, if we wanted to train a neuron model to estimate a specific function, the weights which multiply each input signal will be updated to the output from the neuron is similar to the function.
- Neural networks are composed of elements which operate in parallel. Parallel processing allows increased speed of calculation compared to slower sequential processing.

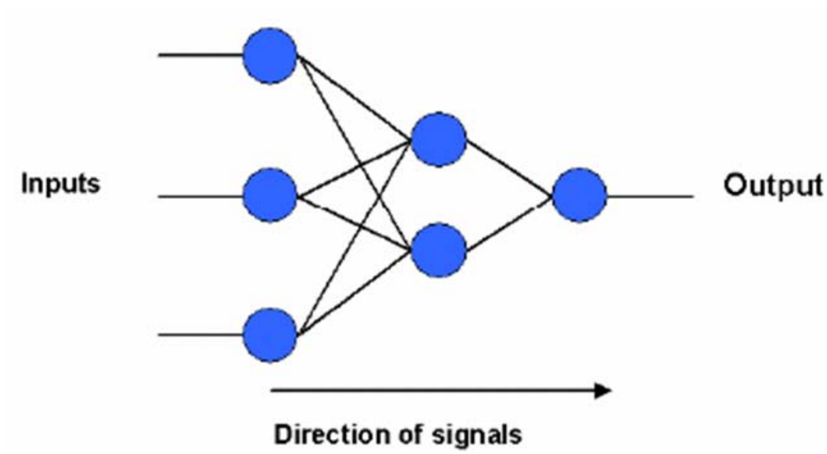


Figure 9: Diagram shows the parallelism of neural networks [15]

- Artificial neural networks (ANN) have memory. The memory in neural networks corresponds to the weights in the neurons. Neural networks are trained offline and then in an adaptive learning process that takes place.

Types of Activation Function **[18: 12-15]**

The activation function defines the output of a neuron in terms of the induced local field υ. There are three basic types of activation functions:

- Threshold function: For this type of activation function, showed in Fugure.10, we have

$$\varphi(\nu) = \begin{cases} 1 & if\,\nu \geq 0 \\ 0 & if\,\nu < 0 \end{cases} \quad (2.5)$$

  In engineering literature, the threshold function is usually referred to as a *Heaviside function*. Correspondingly, the output of neuron k employing a threshold function can be represented by

$$y_k = \begin{cases} 1 & if\,\upsilon_k \geq 0 \\ 0 & if\,\upsilon_k < 0 \end{cases} \quad (2.6)$$

  where υ_k is the induced local field of the neurons; that is,

$$\upsilon_k = \sum_{j=1}^{m} w_{kj} x_j + b_k \quad (2.7)$$

- Piecewise-Linear Function: For the piecewise-linear function showed in Figure. 10, we have

$$\varphi(\nu) = \begin{cases} +1, & \nu \geq +1 \\ \nu, & -1 < \nu < +1 \\ -1, & \nu \leq -1 \end{cases} \quad (2.8)$$

  where the amplification factor inside the linear region is assumed to be unity. This form of an activation function can be regarded as an approximation to a nonlinear amplifier.

- Sigmoid Function: The sigmoid function is the most common form of activation used in the construction of artificial neural network. An example of the sigmoid function is the logistic function, represented by

$$\varphi(v) = \frac{1}{1+\exp(-av)} \tag{2.9}$$

where a is the slope parameter of sigmoid function. By changing the parameter a, we can obtain sigmoid functions of different slopes.

The activation function showed in Eqs. (2.5), (2.8) and (2.9) range from 0 to +1. Having the activation function range from -1 to +1 is desirable, the threshold function of equation (2.5) can be defined as

$$\varphi(v) = \begin{cases} +1 & if\, v > 0 \\ 0 & if\, v = 0 \\ -1 & if\, v < 0 \end{cases} \tag{2.10}$$

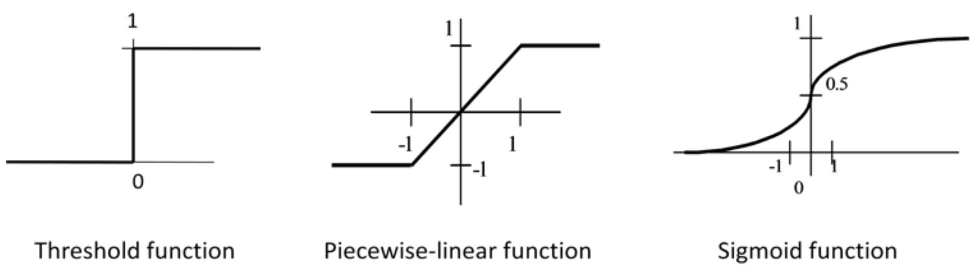


Figure 10: Three types of activation

This research employed a log-sigmoid function.

Dynamic multiresponse system

A dynamic system with multiresponse can be shown as:

$$y_{jk} = f_{jk}(M_k, X) + e_{jk}\text{, for j = 1,2,…,r; k= 1,2,…,s.} \quad (2.11)$$

where f_{jk} is the response function between the control factors and the jth response at the k_{th} level of signal factor; and e_{jk} is a random error. For each dynamic response, a linear form exists between the response and the signal factor. The ideal function can be shown as $y=\beta M + e$, where y denotes the response, M stands for the signal factor, β is the slope or system's sensitivity, and e represents the random error [10]. This research considers two controllable factors with 100 levels each and three noise factors with 2 levels each. Single factor enters into the system, and only one response variable is created by the ANNs.

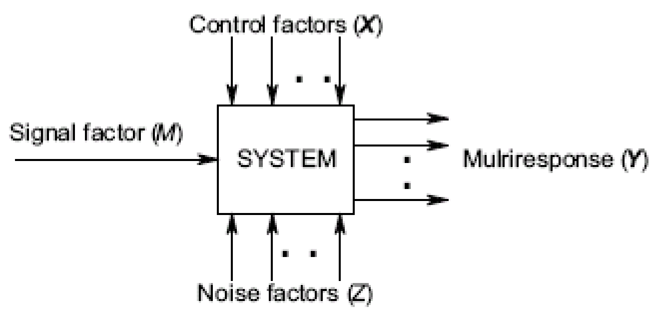


Figure 11: The Parameter Diagram of a dynamic muliresponse system [10]

Linearly Constrained Discrete Optimization (LCDO)

Optimization is an important tool in decision science and in the analysis of systems. In order to make use of this tool, we have to first identify an objective function and its variables. Our goal is to find the optimal threshold combinations that optimize the objective function. However, the variables are often restricted or constrained. In the optimization process, we first need an appropriate model, which has the process of identifying objective, variables and constraints for a given problem [12:2].The model of optimization including variables and constraints will be presented in the next Chapter.

Mathematically, optimization is the minimization or maximization of a function subject to constraints on its variables [12:3]. We generally use the following notation:

1. x is the vector of variables, also called parameters or unknowns;

2. f is the objective function, a (scalar) function of x that we want to maximize or minimize;

3. c_i are constraint functions, which are scalar functions of x that define certain equations and inequalities that the unknown vector x must satisfy [12:3].

Using this notation, the optimization problem can be represented as follows:

$$\min_{x \in R^n} f(x) \quad \text{subject to} \quad \begin{matrix} c_i(x) = 0, i \in E, \\ c_i(x) \geq 0, i \in I \end{matrix} \quad [12:3] \qquad (2.12)$$

Here I and E are the sets of indices for equality and inequality constraints, respectively.

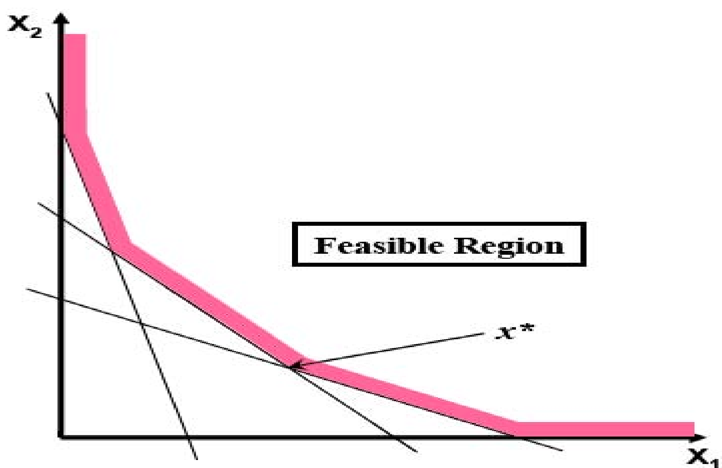


Figure 12: Example of Geometrical Representation of General Optimization Problem [4: 33]

Figure 12 shows the feasible region, which is the set of points satisfying all the constraints, and the point x*, which is the solution of the problem. Sometimes it is more convenient to label the variables with two or three subscripts [12:3-4].

Analysis Techniques

Table 2: Comparison between Kim’s research and this research

	Kim's research	**This research**
Analysis method	Regression Simulation model	Artificial Neural Networks Analytic model
Design	Combined Array	Crossed Array

To evaluate the output data, modelers would employ several techniques of analysis, since it is more advisable than doing just one technique. If the modeler uses one technique, he may get an incorrect evaluation about the output data. In this effort, two different evaluation methods are contrasted. These methods are described in subsequent sections.

Robust Parameter Design (RPD) with Taguchi’s S|N ratio: Crossed Array Design

The RPD is an approach to produce a realization of the activities that emphasizes choice of the levels of controllable factors (or parameters) for two objectives: (1) to ensure that the mean of the output response is at a desired level or target and (2) to ensure that the variability around this target value is as small as possible [11:464]. The original Taguchi methodology for RPD problem revolved around the use of statistical design for the controllable variables and noise variables or uncontrollable variables [11:466]. An indispensible part of the RPD problem is identifying the controllable variables and the uncontrollable variables, and the noise variables affecting the process or product performance, and then finding the optimal settings for the controllable variables that minimize the variability from the noise variables [11:466].

Taguchi's methodology for the RPD problem resolves around the use of orthogonal designs where an orthogonal array involving control variables is crossed with an orthogonal array for the noise variables. For example, in Table 3, the control variables are averaged in a 3^{4-2} factorial design and the noise variables are arrayed in a 2^3 full factorial arrangement. This result is a 72-run design called the crossed array [13].

Table 3: Example of Crossed Array Matrix [11:468].

(a) Inner Array					(b) Outer Array									Responses	
					E	-	-	-	-	+	+	+	+		
					F	-	-	+	+	-	-	+	+		
					G	-	+	-	+	-	+	-	+		
Run	A	B	C	D										mean	SN_L
1	-1	-1	-1	-1		15.6	9.5	16.9	19.9	19.6	19.6	20.0	19.1	17.525	24.025
2	-1	0	0	0		15.0	16.2	19.4	19.2	19.7	19.8	24.2	21.9	19.425	25.522
3	-1	+1	+1	+1		16.3	16.7	19.1	15.6	22.6	18.2	23.3	20.4	19.025	25.335
4	0	-1	0	+1		18.3	17.4	18.9	18.6	21.0	18.9	23.2	24.7	20.125	25.904
5	0	0	+1	-1		19.7	18.6	19.4	25.1	25.6	21.4	27.5	25.3	22.825	26.908
6	0	+1	-1	0		16.2	16.3	20.0	19.8	14.7	19.6	22.5	24.7	19.225	25.326
7	+1	-1	+1	0		16.4	19.1	18.4	23.6	16.8	18.6	24.3	21.6	19.850	25.711
8	+1	0	-1	+1		14.2	15.6	15.1	16.8	17.8	19.6	23.2	24.2	18.313	24.852
9	+1	+1	0	-1		16.1	19.9	19.3	17.3	23.1	22.7	22.6	28.6	21.200	26.152

Taguchi proposed two statistics from the crossed array design: the average of each observation in the inner array for the control variable combination across all runs in the outer array for noise variable combinations, and a summary statistic about the mean and variance, called the *signal-to-noise*(S|N) ratio [11:468]. Then an analysis to decide the setting of the controllable factors is performed for the mean as close as possible to the desired target and a maximum value of the S|N ratio. [11:469]. There are three primary SNRs. The selection of SNRs are depends on the purpose of the experiment; (1) the experimenter wants to achieve a particular target value, (2) the experimenter wants to maximize the response, (3) the experimenter wants to minimize the response [13:540-541]

(1) The target is the best: $SNR_T = -10\log\left(\frac{\overline{y}^2}{S^2}\right)$ (2.13)

(2) The Largest is the best: $SNR_L = -10\log\left(\sum_{i=1}^{n}\frac{1/y_i^2}{n}\right)$ (2.14)

(3) The Smallest is the best: $SNR_S = -10\log\left(\frac{1}{n}\sum_{i=1}^{n}y_i^2\right)$ (2.15)

However, the mean and variance modeling approach using a cross array design has a disadvantage that no direct benefit from the interactions between controllable variables and noise variables, and in some examples, it can even mask these relationships [11:471]. If we think of the SNRs (smallest is best), equation (2. 15), while $\sum y_i^2/n$ is the variability around the target of zero, it is clear that an analysis of the use of this SNR cannot be separated from the location effects due to dispersion effect [13:542]. Thus, it can be shown that

$$(\sum_{i=1}^{n}y_i^2/n) = \overline{y}^2 + \frac{1}{n}(\sum_{i=1}^{n}y_i^2 - n\overline{y}^2) = \overline{y}^2 + (\frac{n-1}{n})S^2 \qquad (2.16)$$

In the following chapter, we use variance instead of SNR, since this research considers mean and variance as the response variables.

Robust Parameter Design: Combined Array Design and the Response Model

Table 4: Example of a Combined Array Matrix [11:476]

Run number	x1	x2	z1	z2	z3	y
1	-1.00	-1.00	-1.00	-1.00	1.00	44.2
2	1.00	-1.00	-1.00	-1.00	-1.00	30.0
3	-1.00	1.00	-1.00	-1.00	-1.00	30.0
4	1.00	1.00	-1.00	-1.00	1.00	35.4
5	-1.00	-1.00	1.00	-1.00	-1.00	49.8
6	1.00	-1.00	1.00	-1.00	1.00	36.3
7	-1.00	1.00	1.00	-1.00	1.00	41.3
8	1.00	1.00	1.00	-1.00	-1.00	31.4
9	-1.00	-1.00	-1.00	1.00	-1.00	43.5
10	1.00	-1.00	-1.00	1.00	1.00	36.1
11	-1.00	1.00	-1.00	1.00	1.00	22.7
12	1.00	1.00	-1.00	1.00	-1.00	16.0
13	-1.00	-1.00	1.00	1.00	1.00	43.2
14	1.00	-1.00	1.00	1.00	-1.00	30.3
15	-1.00	1.00	1.00	1.00	-1.00	30.1
16	1.00	1.00	1.00	1.00	1.00	39.2
17	-2.00	0.00	0.00	0.00	0.00	46.1
18	2.00	0.00	0.00	0.00	0.00	36.1
19	0.00	-2.00	0.00	0.00	0.00	47.4
20	0.00	2.00	0.00	0.00	0.00	31.5
21	0.00	0.00	0.00	0.00	0.00	30.8
22	0.00	0.00	0.00	0.00	0.00	30.7
23	0.00	0.00	0.00	0.00	0.00	31.0

Since interactions between controllable and noise factors are the key to a RPD, Montgomery suggests combined array designs and the response model approach that includes both controllable and noise factors and their interactions[11:471]. Table 4 is an example of the combined array design with two controllable and three noise variables (2^{5-1} with center points). Here x1 and x2 are controllable variables, z1,z2 and z3 are noise variables. The model can be shown in regression form:

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \beta_{ij} x_i x_j + \sum_{i=1}^{r} \gamma_i z_i + \sum_{i=1}^{n} \sum_{j=1}^{r} \delta_{ij} x_i z_j + \varepsilon \quad (2.17)$$

where β s are the control coefficients, γ s are the noise coefficients and δ s are the interaction coefficients. It is very easy to generalize this regression form where $f(x)$ is the part of the model involving only the controllable variables and $h(x,z)$ are the terms involving the main effects of noise factors and the interactions between controllable and noise factors[11:472].

$$y(x,z) = f(x) + h(x,z) + \varepsilon \quad (2.18)$$

$$f(x) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \beta_{ij} x_i x_j \quad (2.19)$$

$$h(x,z) = \sum_{i=1}^{r} \gamma_i z_i + \sum_{i=1}^{n} \sum_{j=1}^{r} \delta_{ij} x_i z_j \quad (2.20)$$

If we assume that the mean of noise variables is zero, then the mean model for response can be shown:

$$E_z[y(x,z)] = f(x) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \beta_{ij} x_i x_j \text{ [11:473]} \quad (2.21)$$

and if the covariance is zero, the variance model for response can be shown:

$$V_z[y(x,z)] = \sum_{i=1}^{r} \left[\frac{\partial y(x,z)}{\partial z_i} \right]^2 \sigma_{z_i}^2 + \sigma^2 \quad \text{[11:473]} \quad (2.22)$$

Contour plots (2D) and surface plots (3D) are typically used for showing mean model and variance model. The object is finding the set of parameters with the highest expected value and the lowest variance [4:37].

III. Methodology

Introduction

This research is organized in two parts. The first part considers Kim's method using theoretical approaches. In order to create the responses, Kim's method uses the ROC analysis and Monte Carlo simulation mentioned in Chapter 2, however Monte Carlo simulation in the Matlab code is complex and requires too much time. Thus, this research replaces Kim's method with analytical techniques based on Bayes's rule.

The second part compares output results between Kim's and ANNs method. Both methods have same CID scenario which is an Air to Ground scenario. The basic concept is shown in the figure below.

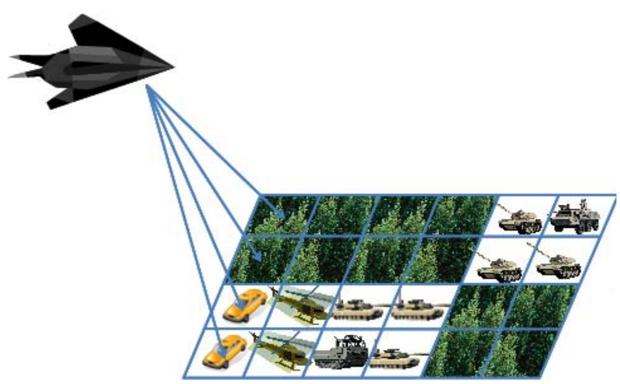


Figure 13: Concept Picture of CID Process [4:38]

First, the friendly force's aircraft divide the ROI into constant size blocks. Then the aircraft performs detection and classification for each block and saves the result as data in the model. In this effort, we assume Non-cooperative communication for doing detection and classification in the given ROI, and declare enemy, friend or clutter based on the output of the system. Kim's method uses the ROC analysis and Monte Carlo simulation mentioned in Chapter 2 to create the responses (the label accuracies) of the simulation, however, this research uses a theoretical method that will be mentioned later. After finding the responses, Kim's method obtains optimal ROC threshold settings by applying RPD with a combined array design. This research also finds optimal ROC settings by using ANNs with a crossed array design. CID simulation needs several inputs, such as: an artificially formed area (battlefield) consisting of enemies, friends, neutrals and clutter, prior confusion matrices (CM) obtained from predetermined ROC curves and cost coefficients associated with the incorrect detection and classification. In this research, the prior ROC threshold is identical to the prior CM because, predetermined ROC thresholds are expressed through the prior CM (See Table 1). The most important output data of the CID simulation is the CM with attributes to obtain optimal ROC thresholds settings which optimize objective functions such as maximum label accuracy of the system and minimum error. [4:38-39]

Validation of Kim's Method

Flow chart of CID

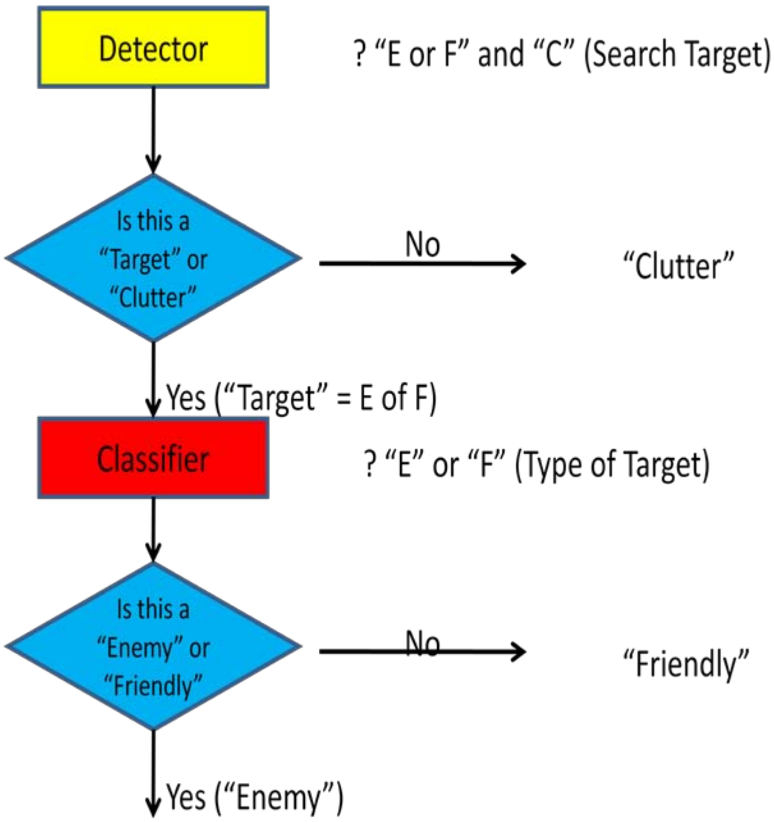


Figure 14: Flow Chart of CID

Figure 14 shows the flow of CID. First, the detector declares a potential target as clutter or possible friendly or enemy. If the target is clutter, it is labeled "C". If it is friendly or enemy, it is passed to a classifier which is then used to discriminate between friendly (F) and enemy (E). After detection, the classifier classifies the data that the detector sent. If the classifier declares it is enemy, then the system recognizes it as enemy. And if the classifier declares it is friendly, then the system recognizes it as friendly.

The TPR, FPR and Label Accuracy

Detector level

Table 5: CM of detector level [4:27]

Detector "Labels"	True Classes: Enemy or Friend	Clutter	Horizontal Totals
"Enemy or Friend"	E or F labeled "EF"	C labeled "EF"	"E or F" Declared
"Clutter"	E or F labeled "C"	C labeled "C"	"C" Declared
Vertical value	E or F evaluated	Clutter evaluated	

$$P(\text{"}E \cup F\text{"} \mid E \cup F) = \frac{P((\text{"}E \cup F\text{"}) \cap (E \cup F))}{P(EUF)} = \frac{\text{first row and column of detector's CM}}{\text{sum of first column of detector's CM}} \quad (3.\ 1)$$

$$P(\text{"}E \cup F\text{"} \mid C) = \frac{P((\text{"}E \cup F\text{"}) \cap C)}{P(C)} = \frac{\text{first row and second column of detector's CM}}{\text{sum of second column of detector's CM}} \quad (3.\ 2)$$

$$P(E \cup F \mid \text{"}E \cup F\text{"}) = \frac{P((E \cup F) \cap (\text{"}E \cup F\text{"}))}{P(\text{"}E \cup F\text{"})} = \frac{\text{first row and column of detector's CM}}{\text{sum of first row of detector's CM}} \quad (3.\ 3)$$

Equation (3. 1) is a TPR, (3. 2) is a FPR and (3. 3) is a Label accuracy of detector level.

Classifier level

Table 6: CM of classifier level [4:27]

Classifier "Labels"	True Classes: Enemy	Friend	Clutter	Horizontal Totals
"Enemy"	E labeled "E"	F labeled "E"	C labeled "E"	"E" Declared
"Friend"	E labeled "F"	F labeled "F"	C labeled "F"	"F" Declared
Vertical value	Enemy evaluated	Friend evaluated	Clutter evaluated	

$$P(\text{"}E\text{"} \mid E) = \frac{P(\text{"}E\text{"} \cap E)}{P(E)} = \frac{\text{first row and column of Classifier's CM}}{\text{sum of first column of Classifier's CM}} \quad (3.\ 4)$$

$$P("E"|F)=\frac{P("E"\cap F)}{P(F)}=\frac{\text{first row and second column of Classifier's CM}}{\text{sum of second column of Classifier's CM}} \quad (3.\ 5)$$

$$P(E|"E")=\frac{P(E\cap "E")}{P("E")}=\frac{\text{first row and column of Classifier's CM}}{\text{sum of first row of Classifier's CM}} \quad (3.\ 6)$$

Equation (3. 4) is a TPR, (3. 5) is a FPR and (3. 6) is a Label accuracy of classifier level.

System level

Table 7: CM of system level [4:27]

	True Classes			
System "Labels"	Enemy	Friend	Clutter	Horizontal Totals
"Enemy"	E labeled "E"	F labeled "E"	C labeled "E"	"E" Declared
"Friend"	E labeled "F"	F labeled "F"	C labeled "F"	"F" Declared
"Clutter"	E labeled "C"	F labeled "C"	C labeled "C"	"C" Declared
Vertical value	Enemy evaluated	Friend evaluated	Clutter evaluated	

$$P("E"|E)=\frac{P("E"\cap E)}{P(E)}=\frac{\text{first row and column of system's CM}}{\text{sum of first column of system's CM}} \quad (3.\ 7)$$

$$P("E"|F)=\frac{P("E"\cap F)}{P(F)}=\frac{\text{first row and second column of system's CM}}{\text{sum of second column of system's CM}} \quad (3.\ 8)$$

$$P(E|"E")=\frac{P(E\cap "E")}{P("E")}=\frac{\text{first row and column of system's CM}}{\text{sum of first row of system's CM}} \quad (3.\ 9)$$

Equation (3. 7) is a TPR, (3. 8) is a FPR and (3. 9) is a Label accuracy of system level.

Assumptions

Each detector and classifier occupies a predetermined ROC curve. A neutral force and civilian are mixed with the clutter. There are three characteristics in a virtual ROI such as an enemy, a friendly force, and clutter. All entities must be declared into one of these three categories, and no entity can be non-declared.

Data and Response Variable

Table 8: Example of Design Matrix

Comb. #	TPR_D	FPR_D	TPR_C	FPR_C	Map size	# of Enemy	# of Friend	Rep.
1	0.4422	0.0005	0.4082	0.0005	15	2	2	1
2	0.4932	0.001	0.4082	0.0005	15	2	2	1
3	0.5694	0.0015	0.4082	0.0005	15	2	2	1
4	0.6098	0.002	0.4082	0.0005	15	2	2	1
5	0.644	0.0025	0.4082	0.0005	15	2	2	1
6	0.674	0.003	0.4082	0.0005	15	2	2	1
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
159,996	1	0.048	0.9667	0.05	75	6	6	2
159.997	1	0.0485	0.9667	0.05	75	6	6	2
159,998	1	0.049	0.9667	0.05	75	6	6	2
159,999	1	0.0495	0.9667	0.05	75	6	6	2
160,000	1	0.05	0.9667	0.05	75	6	6	2

There are controllable factors and noise factors in the design matrix shown in Table 8. The controllable factors are the ROC thresholds combination for detection and classification and noise factors are the size of the ROI represented as the total sum of grid points, the number of enemy targets and the number of friendly targets. We have two controllable factors with 100 levels each and three noise factors with 2 levels each. Also this data has two replications. Thus the experiment is a full factorial design, consisting of 160,000 design points (100 2 * 24 = 160,000) [4:42].

In this section, we have only one response variable. The TPR of the real system defined as P("E"│E),and is generally determined in test environment. In contrast, a warfighter actually does not want the TPR of the system, P("E"│E) but rather P(E│"E");

they want to know the label accuracy of the target of interest to avoid tragedies such as fratricide, collateral damage, and so on before they make decision and firing.[4:52]

Kim's Method [4:43]

Establishment of Virtual ROI to Set up System Environment

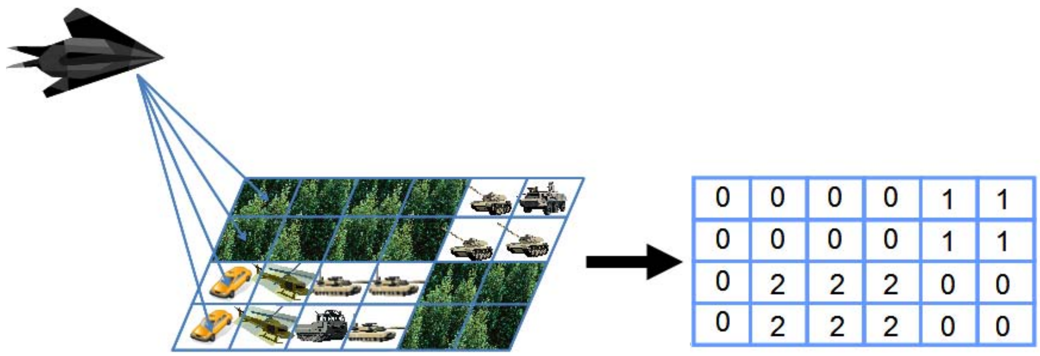


Figure 15: Configuration of ROI

Figure 15 shows the process of configuring a real ROI to virtual ROI via a matrix to execute as a simulation. The CID process requires a virtual ROI to employ given thresholds since detection and classification use a virtual ROI when they evaluate each grid with a specific prior ROC threshold. There are a number of components that construct an actual battlefield; however, this model deals only with enemy, friend, and clutter (clutter includes neutrals, civilians, and all objects other than enemy or friendly). In the virtual ROI, the enemy is represented by"1", friend is represented by "2", and clutter is expressed by "0". Each grid point can only have one characteristic out of three (enemy, friend and clutter). As it is shown at Figure 15, the matrix established by these three figures can be thought as a virtual ROI. Once the virtual ROI is established, the system tests all ROC threshold combinations by comparing it with random numbers and declares the grid point enemy, friend or clutter based on the result of the comparison. The virtual ROI is considered a noise factor because in the case of a real battlefield, the size of the ROI, the characteristics of the grid (enemy, friend or clutter), the number of enemy and that of friend in the ROI, and so forth are generally hard to predict.

Detection and Classification Process [4:46-47]

The model established a virtual ROI according to the design matrix at the opening of the simulation. The system performs detection and classification processes and makes posterior CMs by employing 10,000 prior CM combinations at the established virtual ROI. To test one prior CM combination, Kim uses Monte Carlo simulations, a random number comparison method. That is, the system compares its prior CM combinations with a random number from 0 to 1 in terms of every grid point which is on the pre-established virtual ROI and decides success or failure of the detection and the classification.

```
 for k = 2:numberchoices
          out(k,1) = prob(k) + out(k-1,1);
 end
check = 0;
index = 1;
while check == 0
        if out(index,1) >= rand(1)
            output1(i,j) = column_d(index);
            check = 1;
        else
            index = index + 1;
        end
end
```

True Classes	
Enemy or Friend	Clutter
Prob ("EF" | E or F)	Prob ("EF" | C)
Prob ("C" | E or F)	Prob ("C" | C)

True Classes	
Enemy or Friend	Clutter
Prob ("EF" | E or F)	Prob ("EF" | C)
Prob ("EF" | E or F) + Prob ("C" | E or F) = 1	Prob ("EF" | C) + Prob ("C" | C) = 1

Figure 16: The Part of detection and Classification MATLAB Code and its Description

As we see at ROC curve theory, the sum of TPR and FNR and that of FPR and TNR are equal to 1. The matrix on the top right (a prior CM for detection) of the Figure16 is a graphical representation of first three lines of the MATLAB code on the left while the remaining lines perform transition to lower matrix. The MATLAB function, “Rand (1)” creates a random number between 0 and 1. For example, when there is an object (enemy or a friendly force at) on a grid point of the established virtual ROI and the “Rand (1)” is equal to 0.623, then if the TPR of detection is greater than 0.623, the process recognizes the detection of the object, but if not greater than 0.623 the process declares that grid point as clutter. In case of detection, the situation can always be included within one of both mentioned cases because, “Rand (1)” is smaller than one and the sum of TPR and FNR is always one.

Theoretical Method

Label Accuracy of Detector

If we use Bayes’s rule, the label accuracy of Detector is represented by equation (3.10).

$$P(EF|"EF_D") = \frac{P("EF_D"|EF)*P(EF)}{P("EF_D"|EF)*P(EF)+P("EF_D"|C)*P(C)}$$

$$= \frac{TP_D * P(EF)}{TP_D * P(EF) + FP_D * P(C)} \quad (3.10)$$

Label Accuracy of Classifier

The label accuracy of Classifier is shown in equation (3.11). A value of 0.5 of equation (3.11) means that the probability of a target being enemy or friendly given its designation as clutter is equal, that is, P(“E” | C) and the P(“F” | C) are equal.

$$P(E|"E_C") = \frac{P("E_C"|E)*P(E)}{P("E_C"|E)*P(E)+P("E_C"|F)*P(F)+P("E_C"|C)*P(C)}$$

$$= \frac{TP_C * P(E)}{TP_C * P(E) + FP_C * P(F) + 0.5 * P(C)} \quad (3.11)$$

Label Accuracy of System

In this case, since the two events of Detector and Classifier are independent, the label accuracy of System is the transformed equation (3.12).

$$P(E \mid (''(E \square F)_D '' \square '' E_C ''))$$

$$= \frac{P((''(E \square F)_D '' \square '' E_C '') \mid E) * P(E)}{P((''(E \square F)_D '' \square '' E_C '') \mid E) * P(E) + P((''(E \square F)_D '' \square '' E_C '' \mid F) * P(F) + P((''(E \square F)_D \square E_C '') \mid C) * P(C)}$$

$$= \frac{TP_D * TP_C * P(E)}{TP_D * TP_C * P(E) + TP_D * FP_C * P(F) + FP_D * 0.5 * P(C)} \quad (3.12)$$

Comparison between Kim's Method and ANN Method

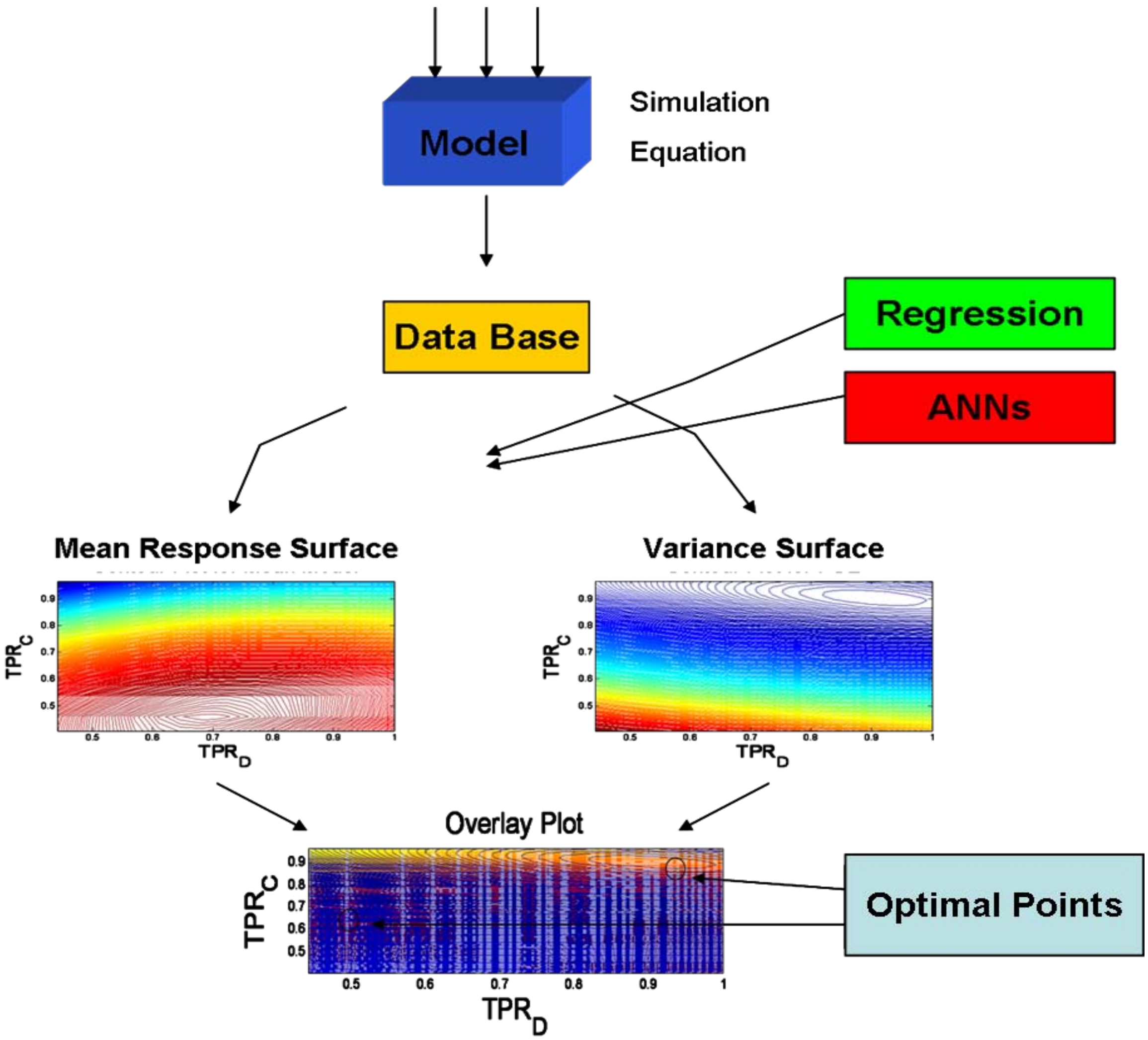


Figure 17: Comparison of Kim's Method and ANN Method

Both methods have similar procedures for the actual experiment. The differences are the model and method for predicted values. Kim's model uses simulation, however, as this research proves, the equation model generates the same response variables. In order to generate predicted values, Kim's method uses regression, while this research uses ANN.

Finding the Feasible Region [4]

After obtaining the responses and other output values, we find the feasible region that satisfies the constraints. Before we determine the feasible region, we need to take an average of system responses for 10,000 different controllable factors (threshold combinations or prior CM combinations). We obtain 6 cases of responses by employing three noise factors with two levels for one specific threshold pair (Detec(FPR, TPR), Class (FPR, TPR)). By taking an average, we can get average values in terms of variance and the system *TPR* for 10,000 different controllable factors. Then we find the feasible region by comparing each average response with its critical value in the following equations.

$$E\text{ (Variance)} \leq \text{maximum Error rate(i)},\ i = 1, 2, 3 \quad (3.13)$$

$$\text{system } TPR \geq \text{ minimum } TPR \quad (3.14)$$

The maximum error rate and the minimum TPR of the system are affected by the quality of ROC curves. This is because if we use low quality ROC curves and high critical values, it is hard to find threshold combinations which satisfy constraints and thus, it is hard to construct a feasible region.

Finding Optimal Threshold Combination

Most decision makers on a real battlefield would want the higher label accuracy and the lower propagation of error (POE). This is because a higher POE could cause unpredicted collateral damage, and lower label accuracy could lead to fratricide in real battlefields. This research finds an optimal threshold combination with the higher mean value and the lowest variance for these variables.

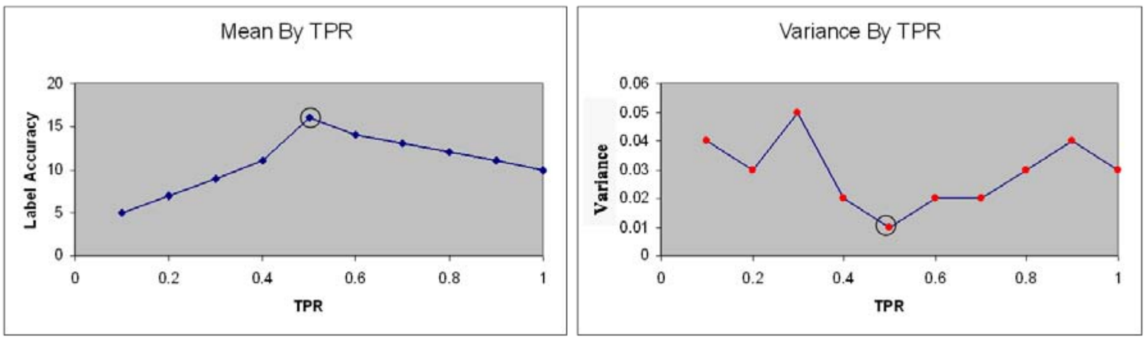


Figure 18: Example of Optimal threshold combination

We can see the optimal point from Figure 18. The 0.5 of TPR has the highest mean value and the lowest variance. However, the highest mean value could also have high variance. In this case, the decision maker should decide optimal threshold combination that has high mean value and appropriate variance in the system.

Evaluation of Output between Kim's method and ANNs

The evaluation methods were briefly explained previously. In this part we consider again the meaning of three measures of performance ($P(E|"E")$, $P(F|"F")$ and $P(C|"C")$), and this research will compare output data between Kim's method and ANNs method.

The residual values are $e_i = y_i - \hat{y}_i$, where $\hat{y}_i$ is the predicted or fitted value from ANN and regression analysis. Residuals provide considerable information about unexplained variability. [13: Sec 2, 7] For example, when the range of residuals is wide, the unexplained variance is also high.

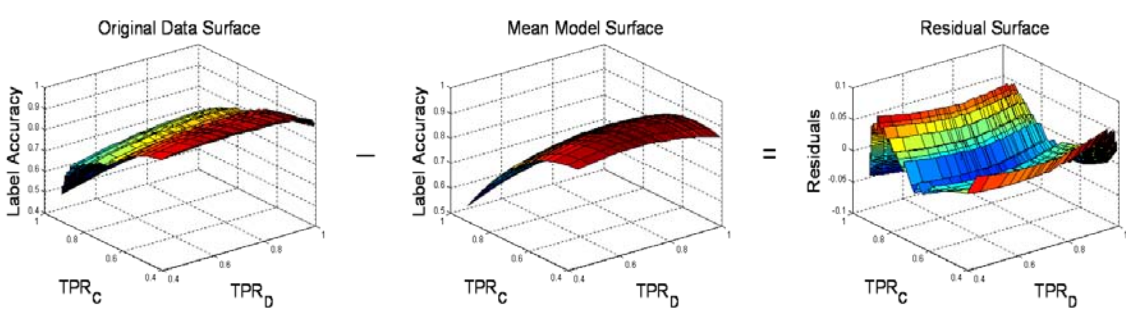


Figure 19: Example of residual in CID

***RPD with Combined Array Design* [4]**

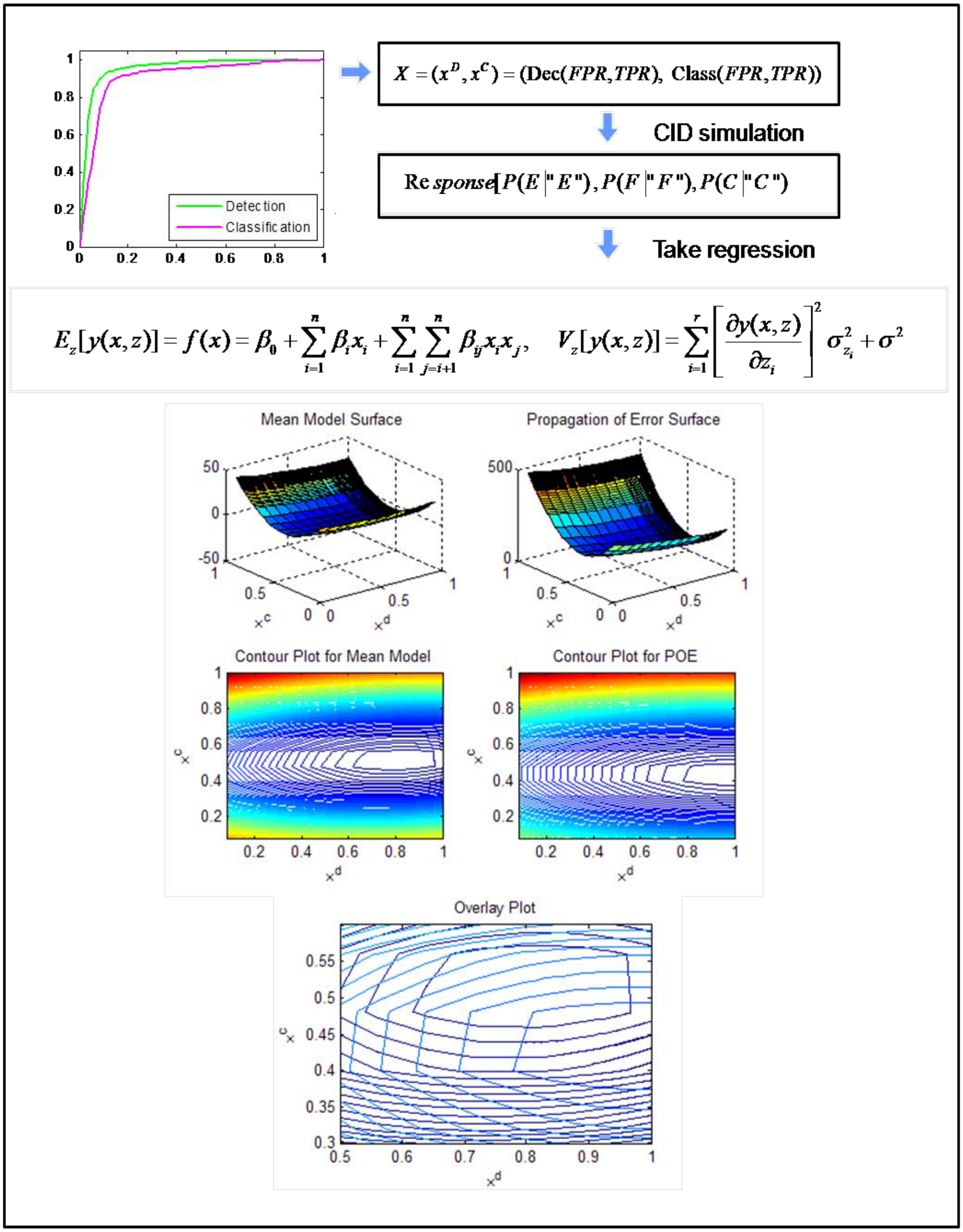


Figure 20: CID Evaluation Example at RPD

Figure 20 shows the procedure of evaluation by RPD with a combined array design matrix. After finishing the simulation for all threshold combinations, we first do a regression with combined array design, and make a mean model and a propagation of error model. Then the contour plots for those models are constructed and an overlapping figure is also made. By comparing the value of the mean and the propagation error we can find subjective robust point(s).

There is an implicit optimization, that is

$$MAX\ E(\text{Response}(x^D, x^C)) = (\text{Detector}(FPR, TPR), \text{Classifier}(FPR, TPR))$$

Such that

$$VAR(\text{Response}(x^D, x^C)) \leq C$$

ANNs with Crossed Array Design

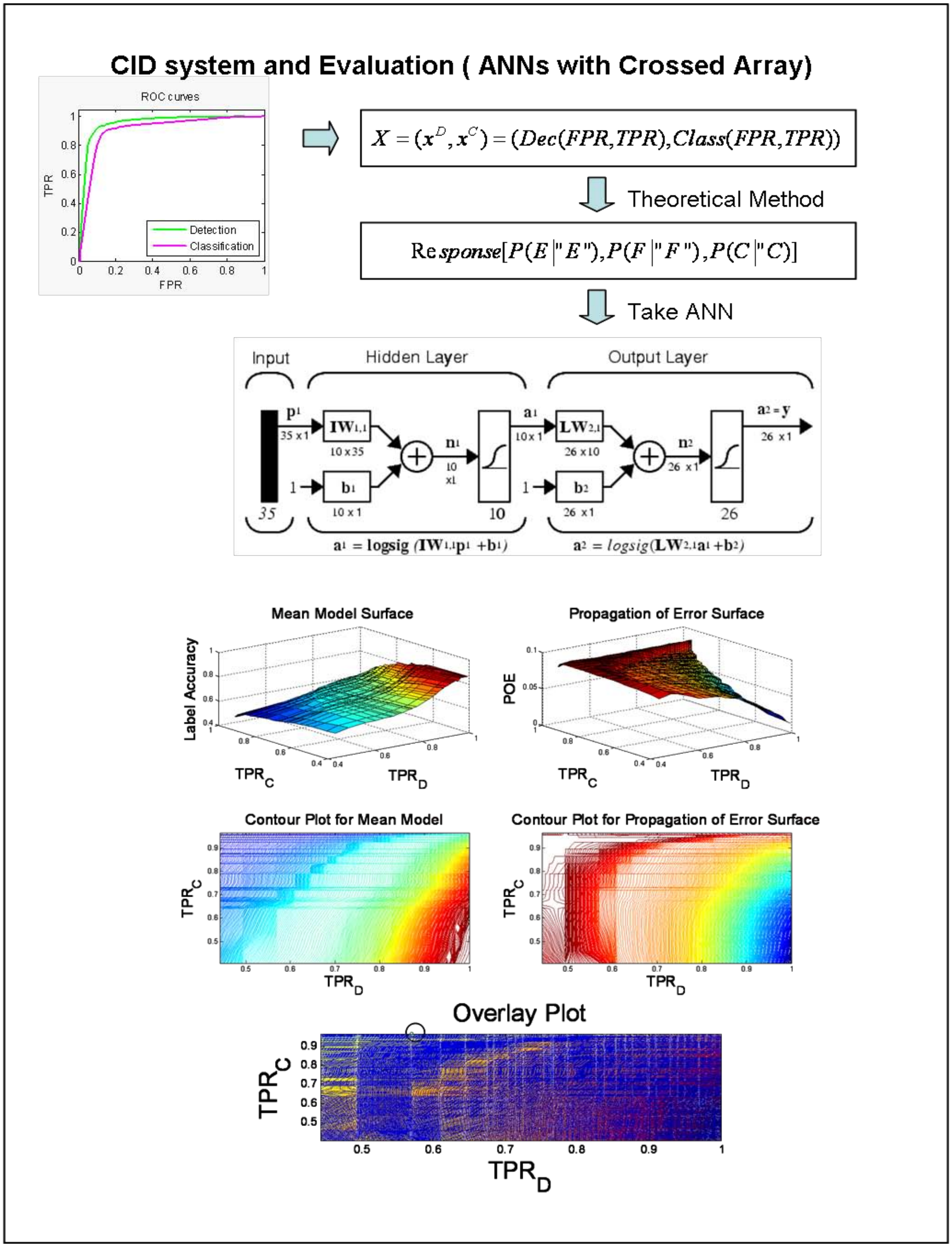


Figure 21: CID Evaluation Example at ANNs

Figure 21 shows the procedure of evaluation by ANNs with a crossed array design matrix. After calculating the equations for all threshold combinations, we first input response variables with crossed array design in the ANNs and make a mean model and a propagation of error model. Then the contour plots for those models are constructed and an overlapping figure is also made. By comparing the value of the mean and the propagation error we can find subjective robust points. [4]

There is an implicit optimization, that is

$$MAX\ E(\mathrm{Response}(x^{D},x^{C}))=(\mathrm{Detector}(FPR,TPR),\ \mathrm{Classifier}(FPR,TPR))$$

Such that

$$VAR(\mathrm{Response}(x^{D},x^{C}))\leq C$$

IV. Experiments and Results

Introduction

Herein are discussed two different sets of experiment and results. The first set compares response variables between Kim's and the analytic method. The second set compares model performance (expected value and variance) between Kim's method and the ANN method. In the second set of experiments for both methods are across the same data sets:

(1) Two notional ROC curves of the detector and classifier. The detector is assumed to perform marginally better than the classifier.

(2) Greatly improved versions of the two notional ROC curves.

As we know through the previous Chapters, we have two responses, these are measures of performance:

(1) Label accuracy for enemy ($P(E|"E")$)

(2) Label accuracy for friend ($P(F|"F")$)

For each ROC curve set, we will get these measures of performance (MoPs) and optimal threshold combinations. In order to generate MoPs, Kim's method uses combined array design and ANN method uses crossed array design.

Analytic Verification of Kim's Method

Output of label accuracy

The Table 9 shows that the mean value of each method is almost same. Additionally, the mean value is also same when the settings of noise variables (map size, number of enemy and number of friendly) are changed. For example, in Table 9, the outputs in the Lim for map size(15) are averaged across the # of enemy and # of friendly.

Table 9: Output data of each Method

	Simulation (Kim)	Theoretical (Lim)
MAP Size(15)	0.932313268	0.93195902
MAP Size(75)	0.755909133	0.755332442
	Simulation (Kim)	**Theoretical (Lim)**
# of ENEMY(2)	0.776789588	0.775971831
# of ENEMY(6)	0.911432814	0.911311631
	Simulation (Kim)	**Theoretical (Lim)**
# of FRIENDLY(2)	0.849920168	0.849472896
# of FRIENDLY(6)	0.838302233	0.837818566
Overall Average	0.844111201	0.843645731

Mean Model Surface

Figure 22 shows the mean model surface plot of each method. The X-axis is a true positive rate of Detector, Y-axis is a true positive rate of Classifier and Z-axis is label accuracy. Both methods have same plots and label accuracies are high when true positive rates of Classifier are between 0.5 and 0.6.

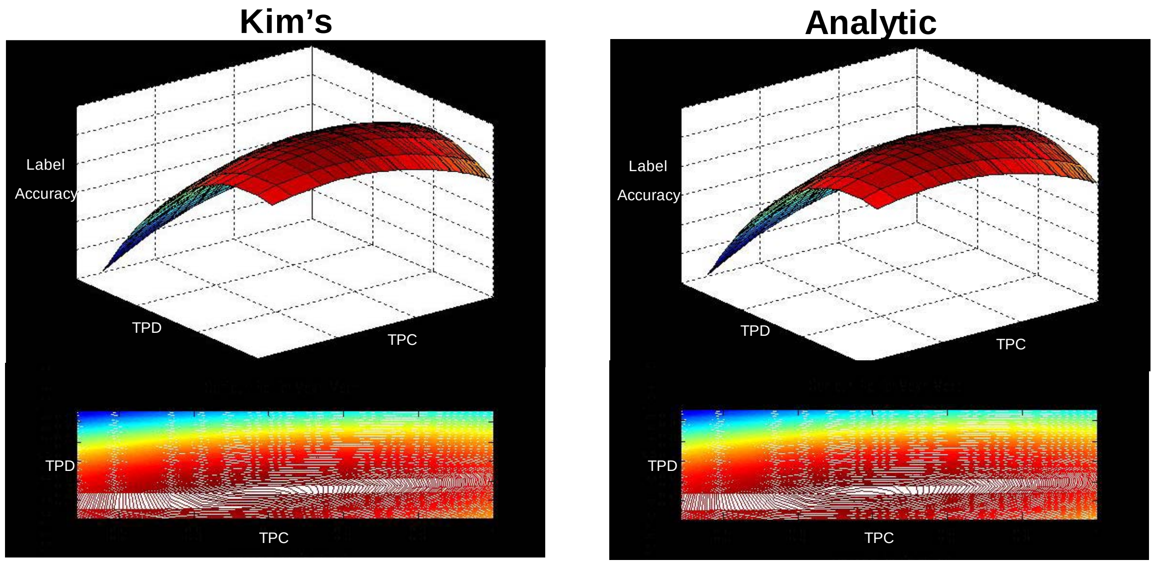


Figure 22: Mean Model Surface Plots

Variance Surface

Figure 23 shows the variance surface plots for each method. The higher variance causes an error on the system. Thus, we want the low variances which are distributed around true positive rate (0.8) of Classifier. Like the mean model surface plot, the two plots are almost the same.

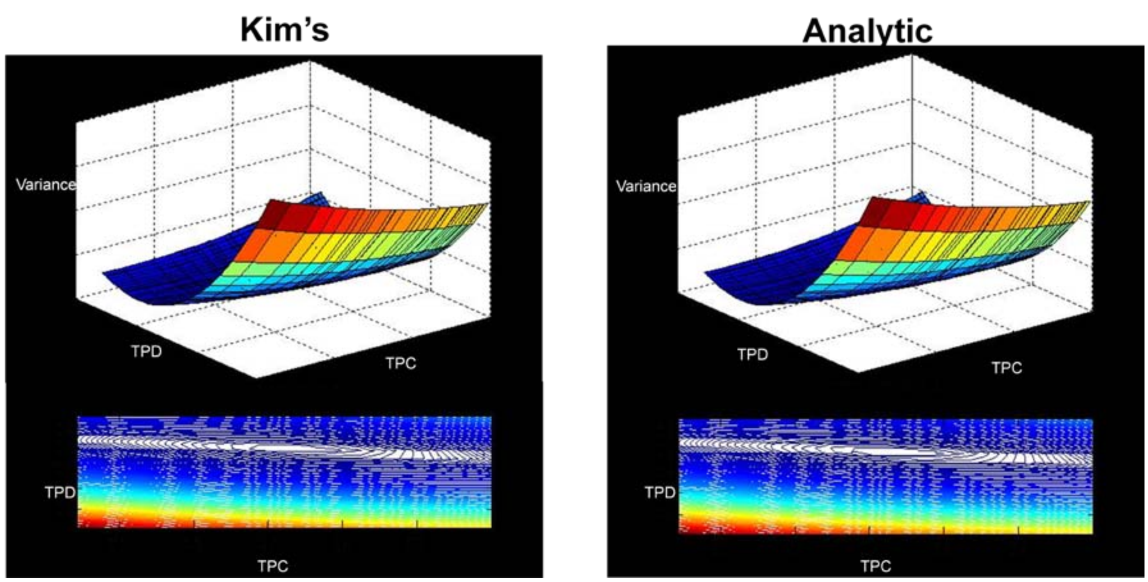


Figure 23: Variance Surface Plots

Comparison output of 1st ROC curve set

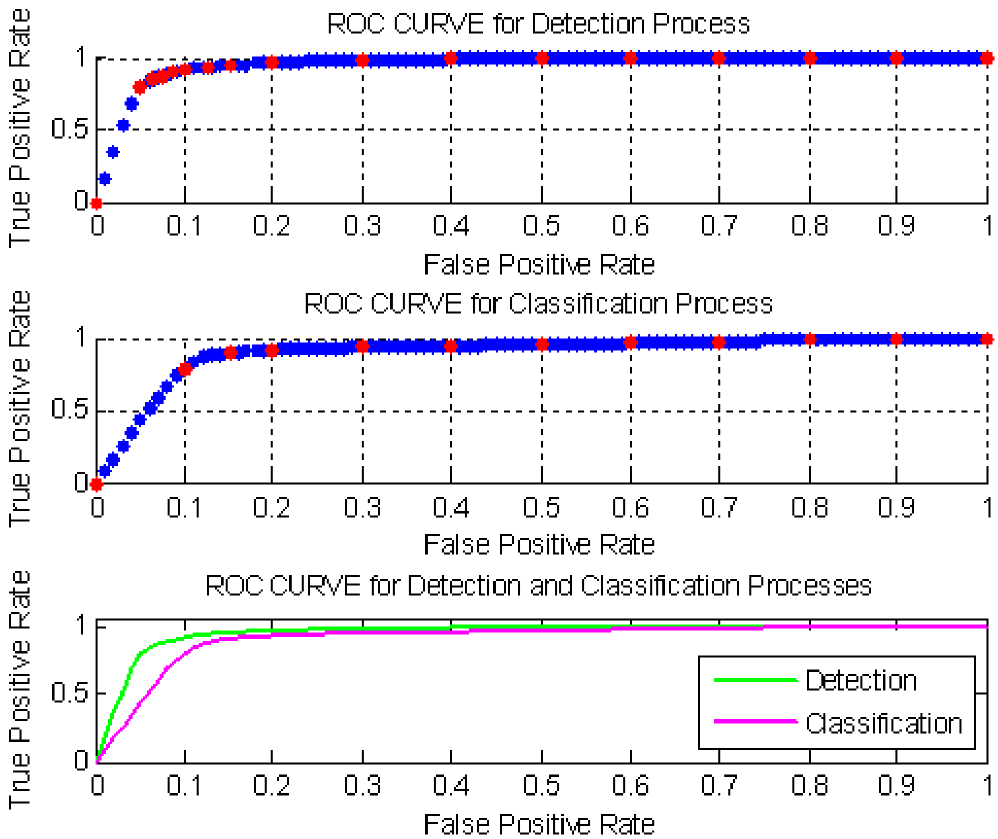


Figure 24: ROC Curves for 1st Experiment Set

These ROC curves are created by RBFs. The red points at the first two graphs have been utilized to erect two ROC curves. From the ROC curves, we gather one hundred pairs of ((FPR), (TPR)) for detection and classification and thus, the total number of ROC threshold combinations is 10,000 [4]. These are two notional ROC curves of the detector and classifier. The detector is assumed to perform marginally better than the classifier.

Label accuracy of Enemy (** $P(E|"E")$ **)

Kim's Method

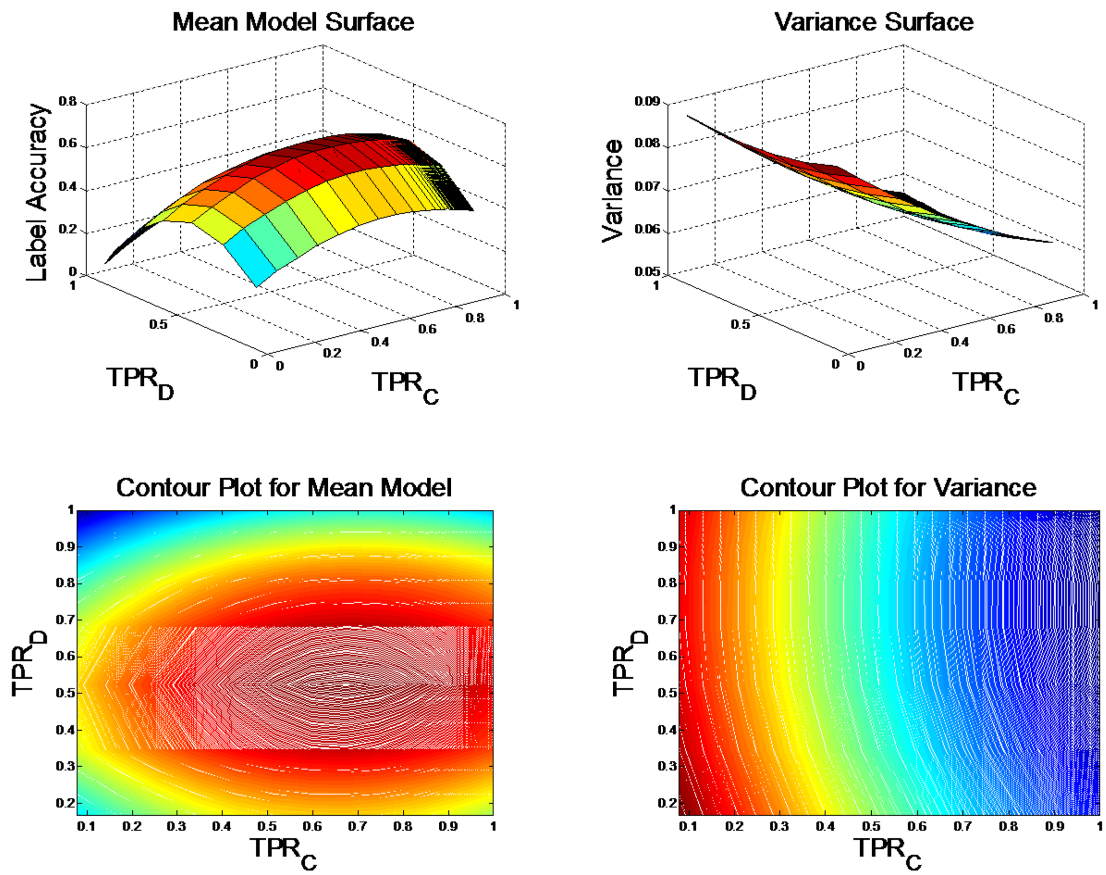


Figure 25: Surface, Contour Plots for Using the Label Accuracy of Enemy for 1st ROC Set

As shown by Figure 25, the highest label accuracy happens when TPR_D is around 0.5 and TPR_C is around 0.65, and the lower variance occurs at the east quadrant of the variance model. We need higher expected value and lower variance, however, the maximum value of label accuracy is poor, because the 1st ROC set has high FPR. In this research, we will employ ANNs in order to capture any non-linear effects missed in Kim's approach.

ANNs Method

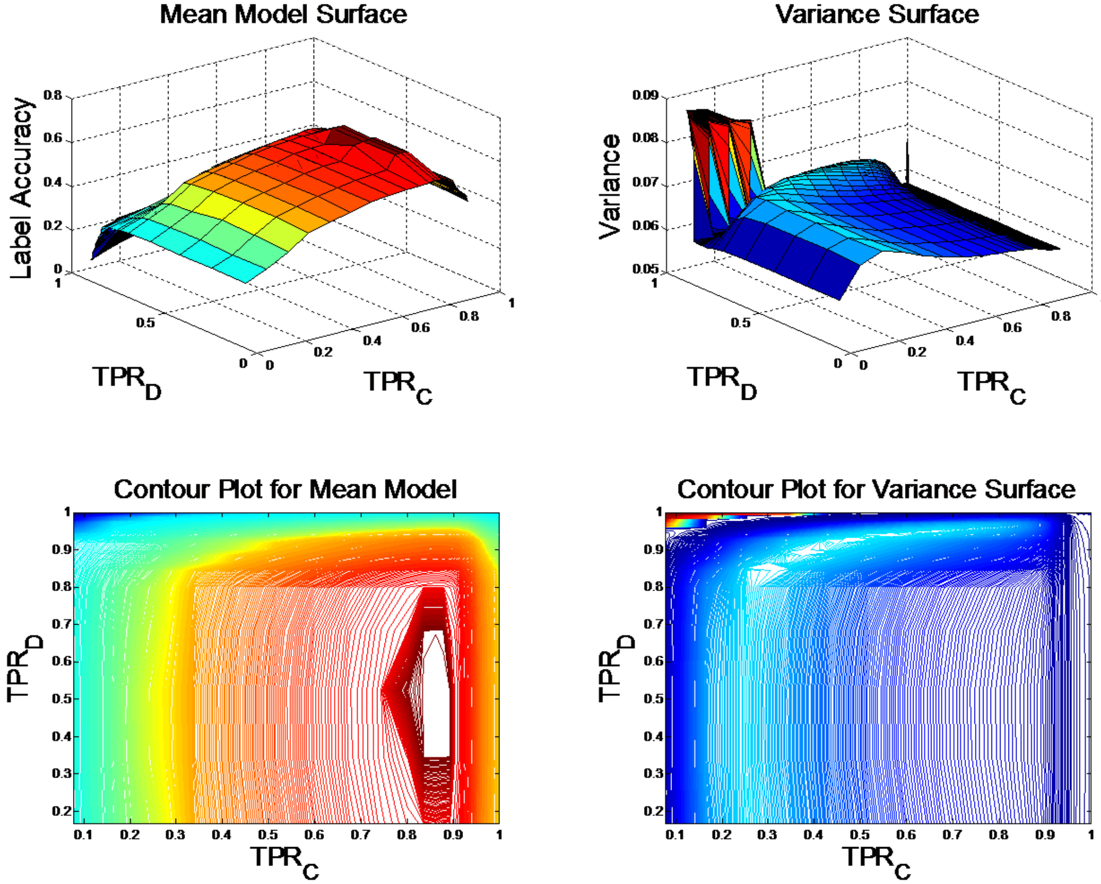


Figure 26: Surface, Contour Plots for Using the Label Accuracy of Enemy for 1st ROC Set

Figure 26 shows more complex expected value and variance than Kim's plots. The higher label accuracy happens when TPR_D is around 0.5 and TPR_C is around 0.85, and the lower variance turns out at the southeast quadrant of the variance model. Like Kim's method, the value of maximum expected value is poor. Seeing the same solution suggests that a poor solution is the best we can expect given the relatively poor ROC curves.

Comparison between Kim's method and ANNs Method

In order to evaluate the ANN method, we compare residual plots between Kim's method and the ANN method. These plots show that Kim's residuals are distributed with greater variance as compared with the ANN method.

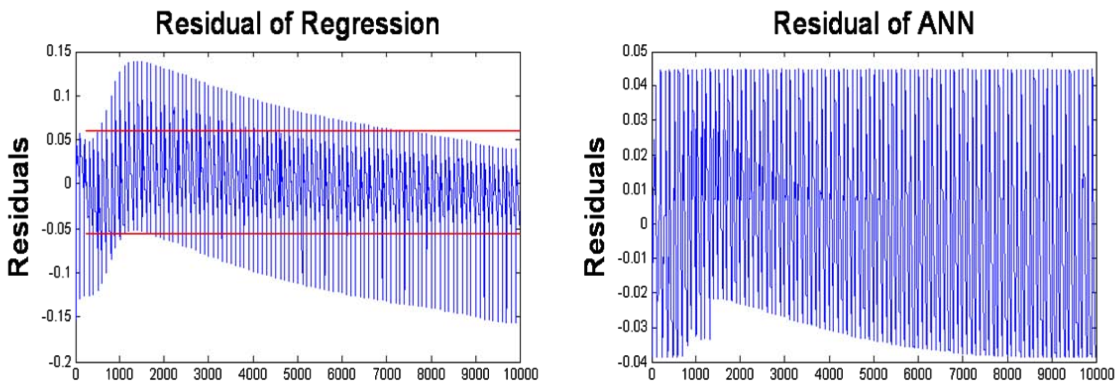


Figure 27: Residual plots of Kim's method and the ANN method (Note scale)

Optimal Points

Though both outputs of expected label accuracies are poor, we are interested in points where we see higher expected value and the lower variance. However, it is difficult to determine the optimal points from surface and contour plots. Thus, this research uses plots of average mean and variance by TPR_D and TPR_C, and mean by variance, in order to confirm optimal point.

Kim's method

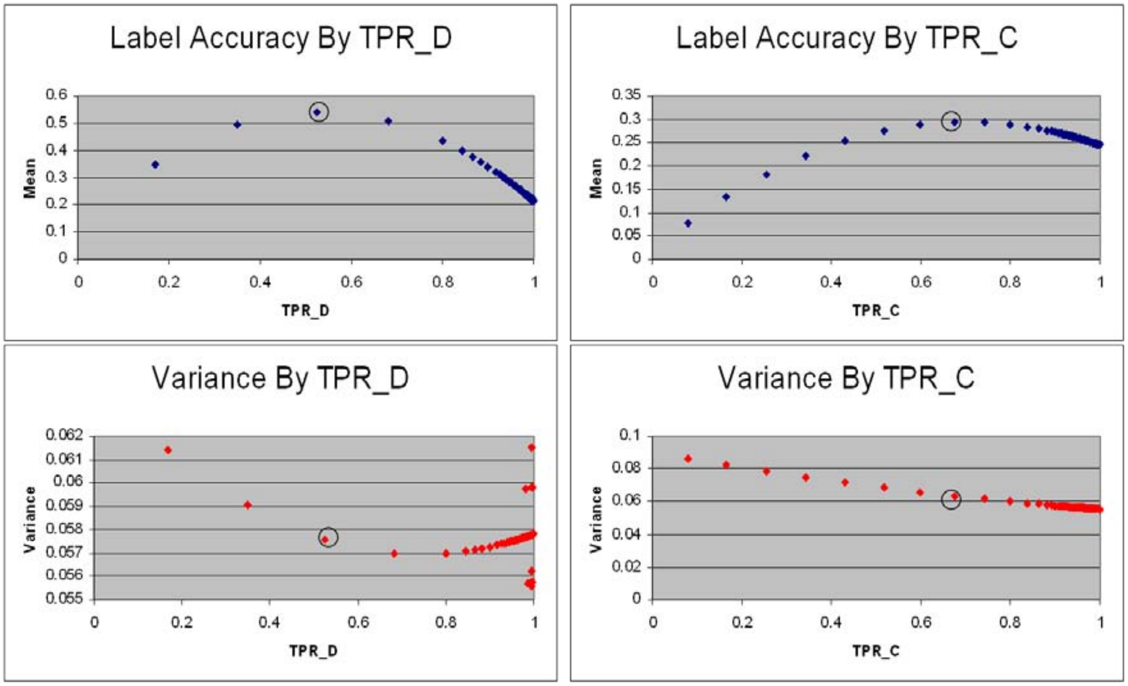


Figure 28: Average Mean and Variance by TPR_D and TPR_C (Kim's Method)

These plots are averaged across all settings. For instance, the circled points on TPR_D settings are average across all TPR_C settings. Figure 28 shows that mean and variance has a negative relation, thus we can determine the best point more easily. The left upper plot indicates the highest TPR_D has a wide range of variance, since the highest TPR_D also has the highest FPR. The optimal point takes place at the black circle that TPR_D is 0.524 and TPR_C is 0.6751.

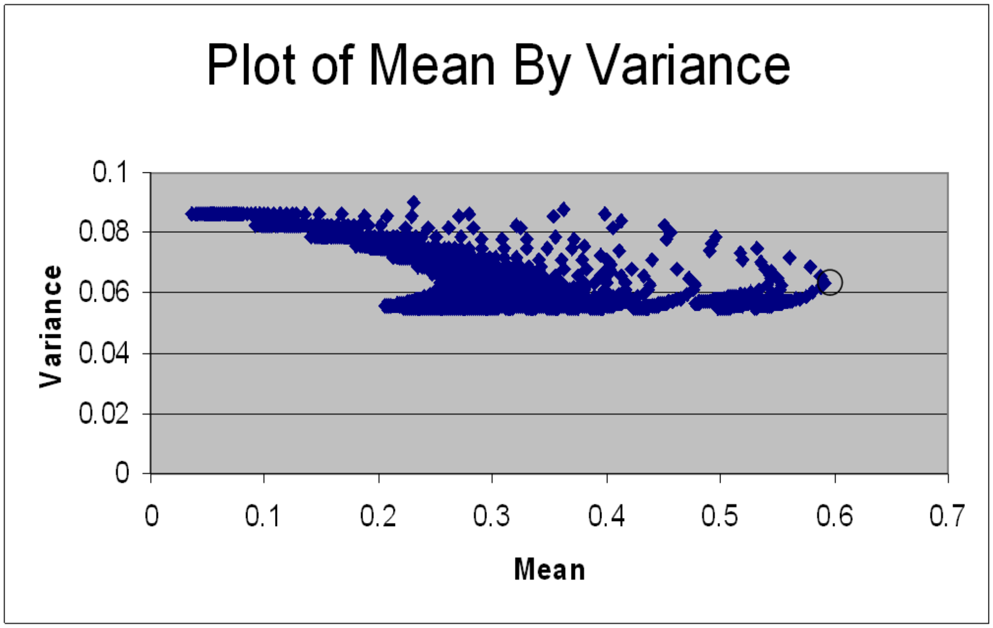


Figure 29: Plot of Mean by Variance

The plot of mean by variance in Figure 29 shows the same optimal point, that is, circled point gives same threshold combination that TPR_D is 0.524 and TPR_C is 6751.

ANNs method

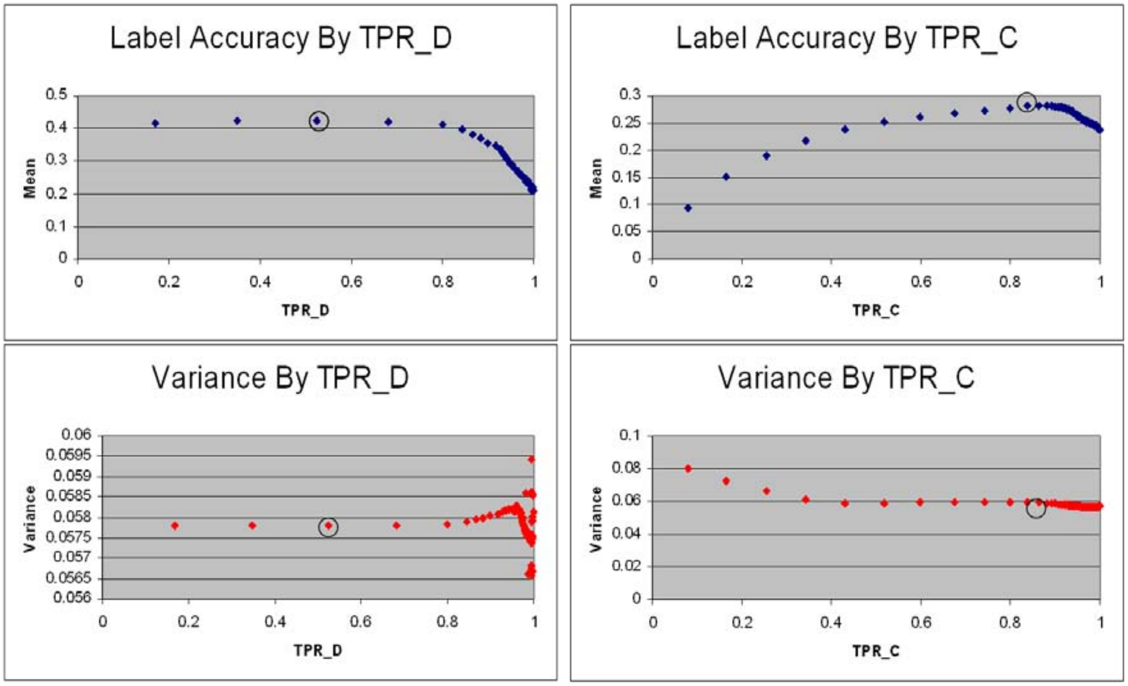


Figure 30: Average Mean and Variance by TPR_D and TPR_C (ANNs Method)

Like Kim's method, the highest label accuracies by TPR_D occur where TPR_D is 0.524, and the range of variance also higher when TPR_D is around 1.0. However, the highest label accuracy is where TPR_C is 0.8919. The optimal point takes place at the black circle that TPR_D is 0.524 and TPR_C is 0.8919. The plot mean by variance in Figure 31 suggests the same optimal point, and makes a clear visual choice.

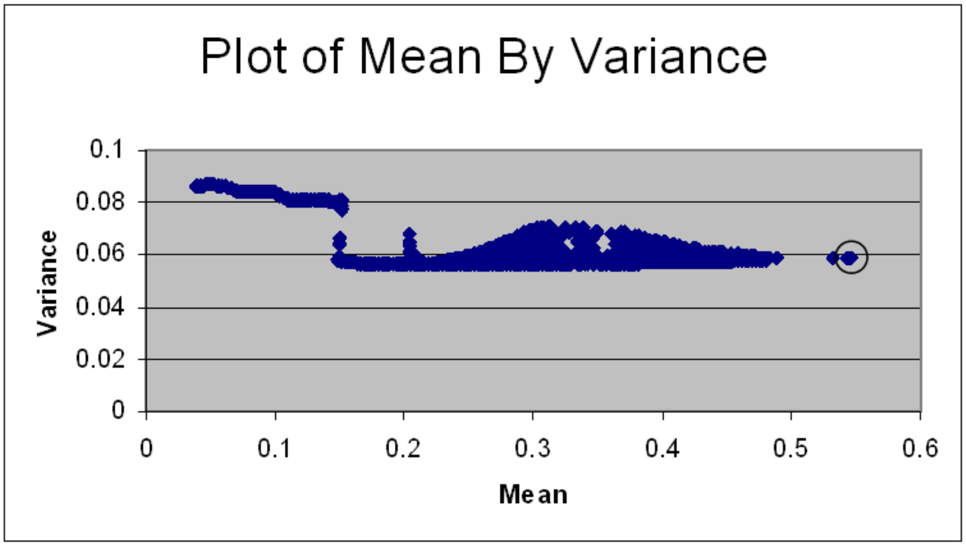


Figure 31: Plot of Mean by Variance

The solutions of the Kim's method are the points of the 703rd threshold combination, and the ANNs method is point of the 1303rd combination. Table 10 shows Kim's solutions have a higher mean value, also have a higher variance.

Table 10: Solution of both Method

Method	Comb#	TPR_D	TPR_C	FPR_D	FPR_C	Mean	Variance
Kim	703	0.524	0.6751	0.03	0.08	0.591509	0.063446
ANN	1303	0.524	0.8919	0.03	0.14	0.544941	0.058444

Label accuracy of Friendly ($P(F\,|\,"F")$ ***)***

Kim's Method

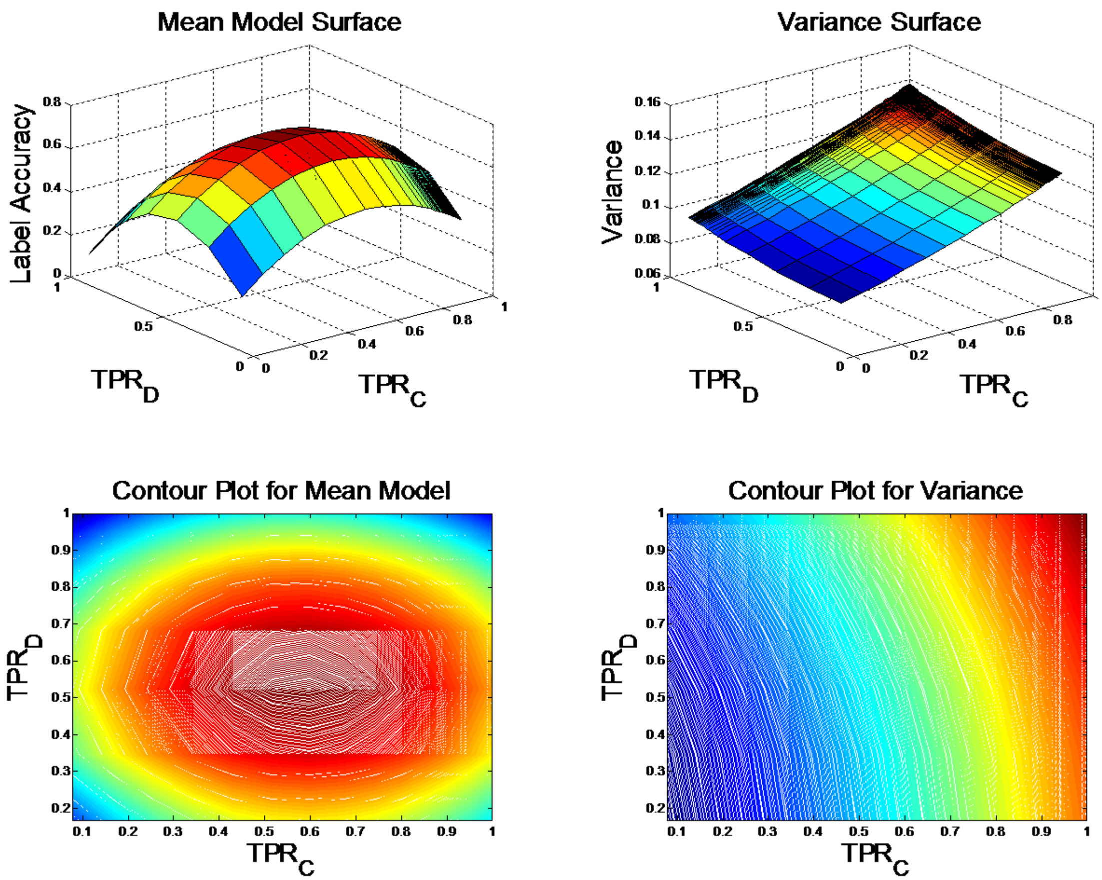


Figure 32: Surface, Contour Plots for Using the Label Accuracy of Friendly for 1st ROC Set

As shown by Figure 32, the highest label accuracy happens when TPR_D is around 0.5 and TPR_C is around 0.6, and the lowest variance occurs at the southwest quadrant of the variance model. The maximum expected value is poor again. We obviously need a better expected value and lower variance, although output seems to indicate a positive relation between label accuracy and variance. Also, we can expect again that ANN would be more accurate.

ANNs Method

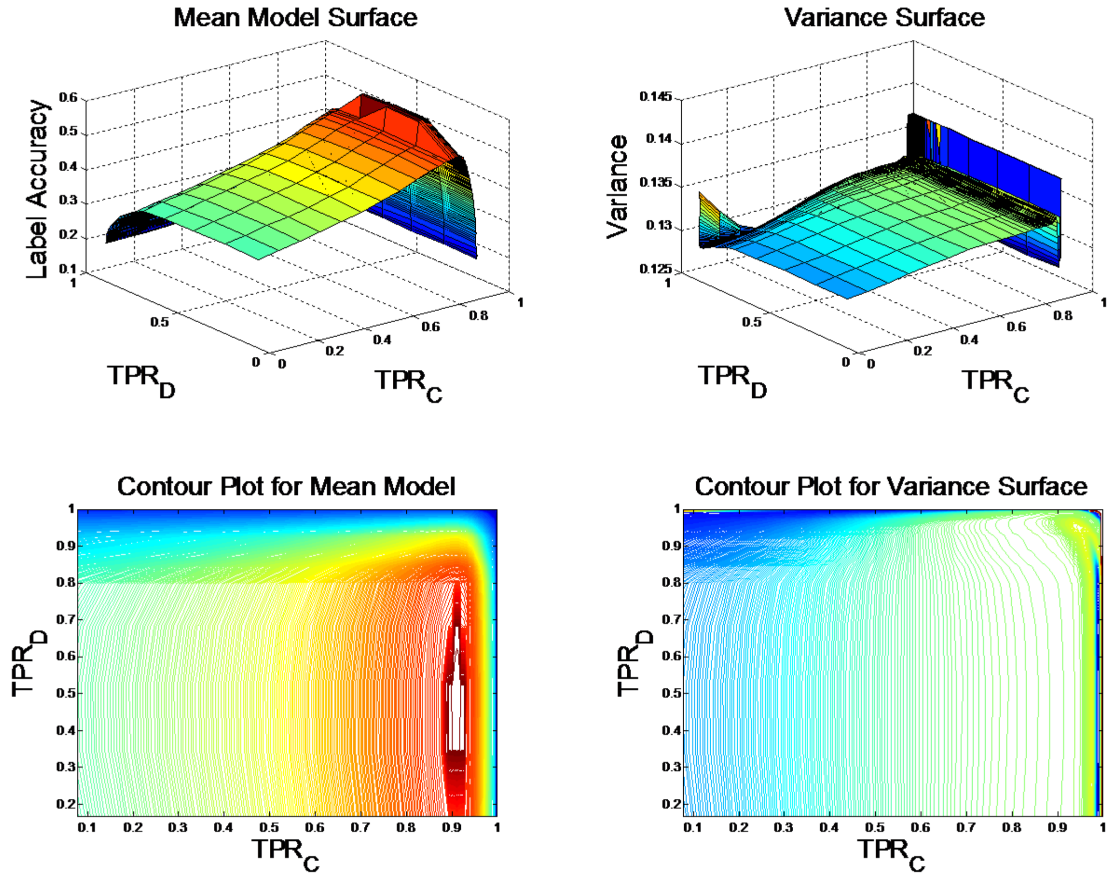


Figure 33: Surface, Contour Plots for Using the Label Accuracy of Friendly for 1st ROC Set

Like plots of label accuracy for enemy, Figure 33 is more complex than plots of Kim's method. Figure 33 shows that the higher label accuracy occurs when TPR_D is around 0.5, and TPR_C is around 0.9. The lower variances are distributed in the east quadrant of model. The maximum label accuracy is poor.

Comparison between Kim's method and ANNs Method

In order to evaluate ANNs method, this research compares again residual plot between Kim's method and ANNs method. These plots show that the residuals of Kim's are wider than the ANN method and the residuals are much greater for friendly label accuracy than for enemy label accuracy.

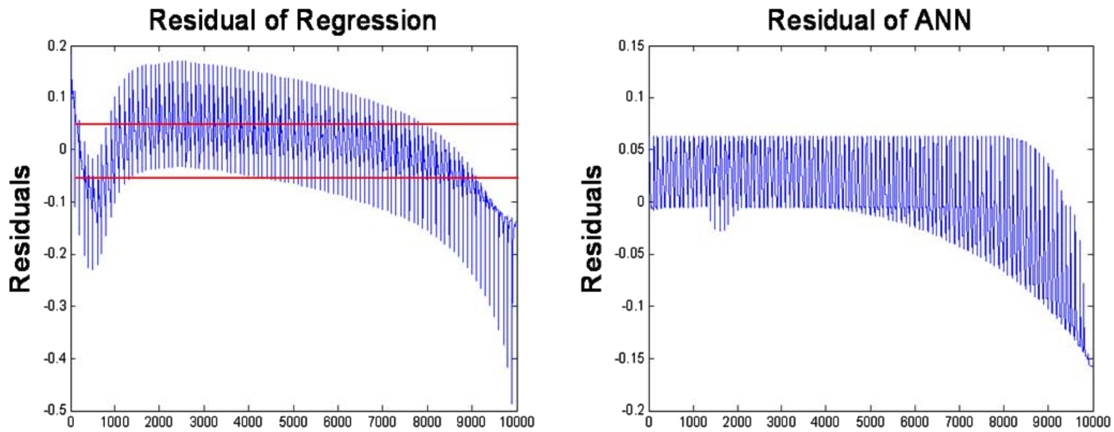


Figure 34: Residual plots of Kim’s method and the ANN method (Note scales)

Optimal Points

It is difficult to confirm optimal points from surface and contour plots. Thus, this research uses plots of average mean and variance by TPR_D and TPR_C, and mean by variance, in order to confirm optimal point.

Kim's Method

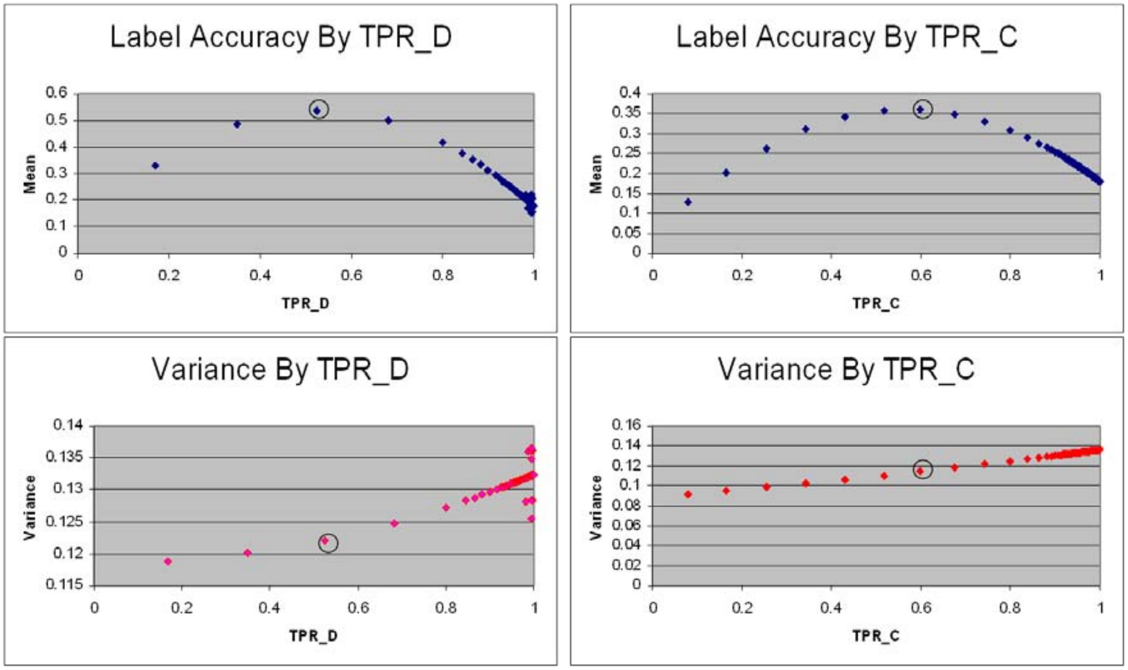


Figure 35: Average Mean and Variance by TPR_D and TPR_C (Kim's Method)

Again the circled points mean averaged across TPR_C and TPR_D. Figure 35 shows that label accuracies and variance have positive relation until middle of TPR_D and TPR_C. The optimal point takes place at the black circle, where TPR_D is 0.524 and TPR_C is 0.5987. The plot of mean by variance in Figure 36 also gives a clear optimal point.

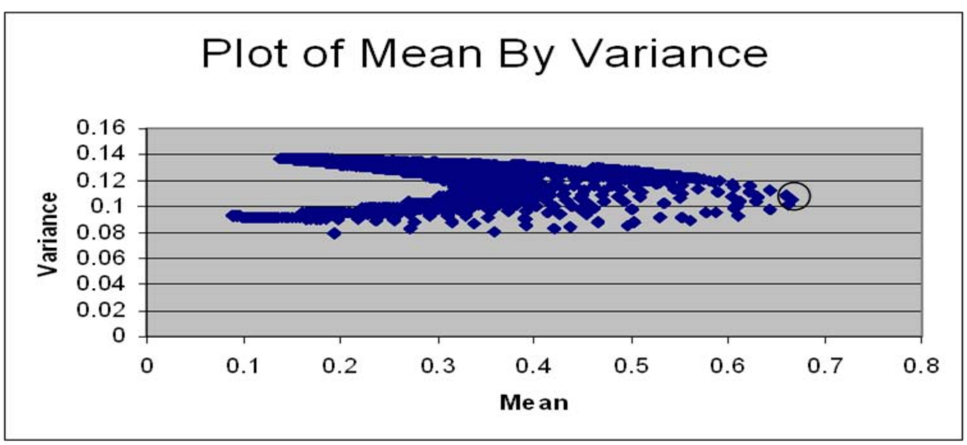


Figure 35: Plot of Mean by Variance

ANNs Method

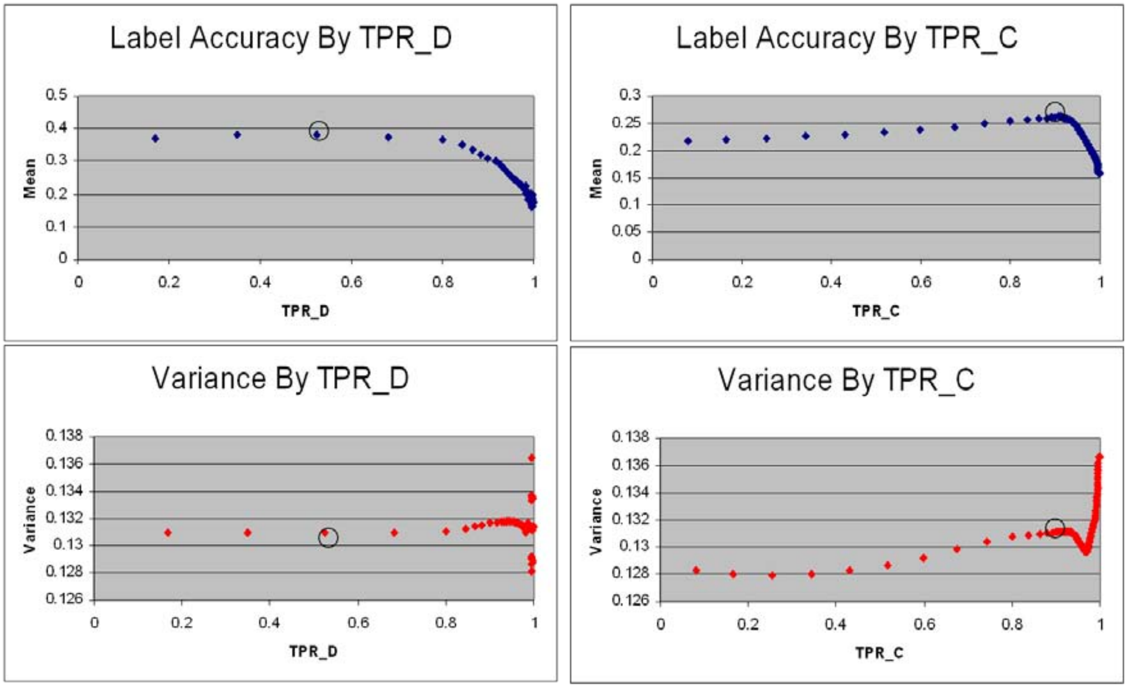


Figure 37: Average Mean and Variance by TPR_D and TPR_C (ANNs Method)

Like Kim's method, the highest label accuracies occur where TPR_D is 0.524, however TPR_C is moved to the right. Thus, the optimal point takes place at the black circle where TPR_D is 0.524 and TPR_C is 0.9067. The plot of mean by variance in Figure 38 shows the same optimal point.

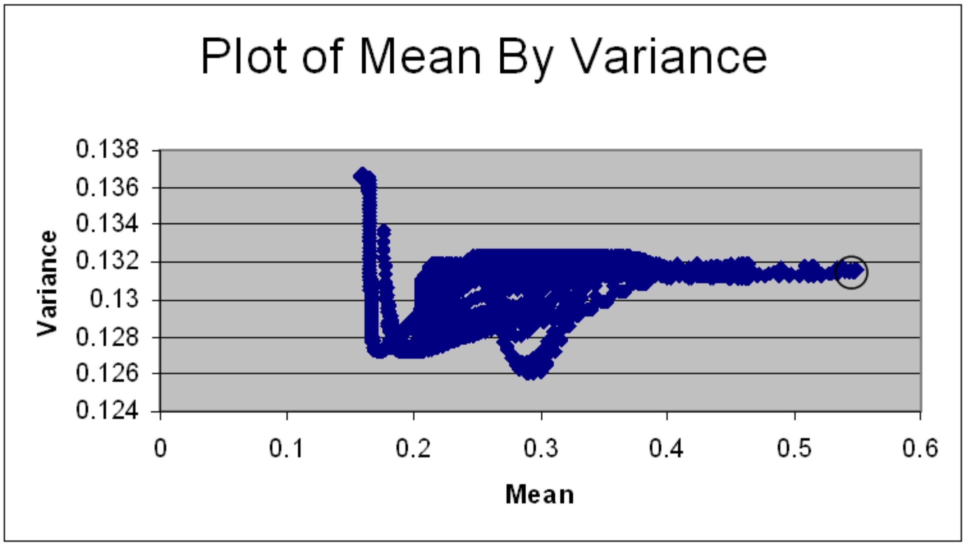


Figure 38: Plot of Mean by Variance

The solutions of Kim's method are the 603rd combination, and the ANNs method is the 1503rd combination. Table 11 shows Kim's solutions have a higher mean value and lower variance.

Table 11: Solution of both Method

Method	Comb#	TPR_D	TPR_C	FPR_D	FPR_C	Mean	Variance
Kim	603	0.524	0.5987	0.03	0.07	0.666694	0.105242
ANN	1503	0.524	0.9067	0.03	0.16	0.547768	0.131611

Comparison output of 2nd ROC curve set

The ROC curves for the CID system are generally determined by the quality of signals and the selection of the decision threshold [14]. If the 1st set of ROC curves has a low quality of signal and hence the region of intersection between the target probability distribution and the clutter probability distribution in the case of detector is relatively large, the 2nd ROC curve set comes up with high quality of signals. Thus, we can expect improved ROC curve behaviors and those are demonstrated at Figure 39[4].

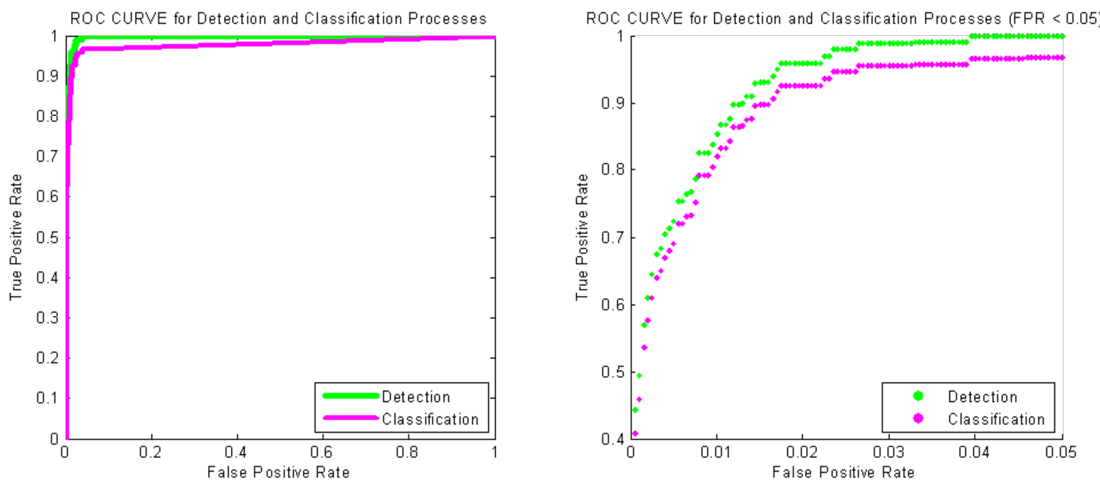


Figure 38: ROC Curves for 2nd Experiment Set

As you see, the ROC curves for 2nd set are much better than previous ones in terms of their high TPR at the same FPR. Right-hand side graph of Figure 39 is used for this experiment and its range of x-axis (FPR) is (0, .05) for both curves. Due to different ROC curves we may see very different results as compared with the 1st ROC set [4].

Label accuracy of Enemy ($P(E|"E")$***)***

Kim's Method

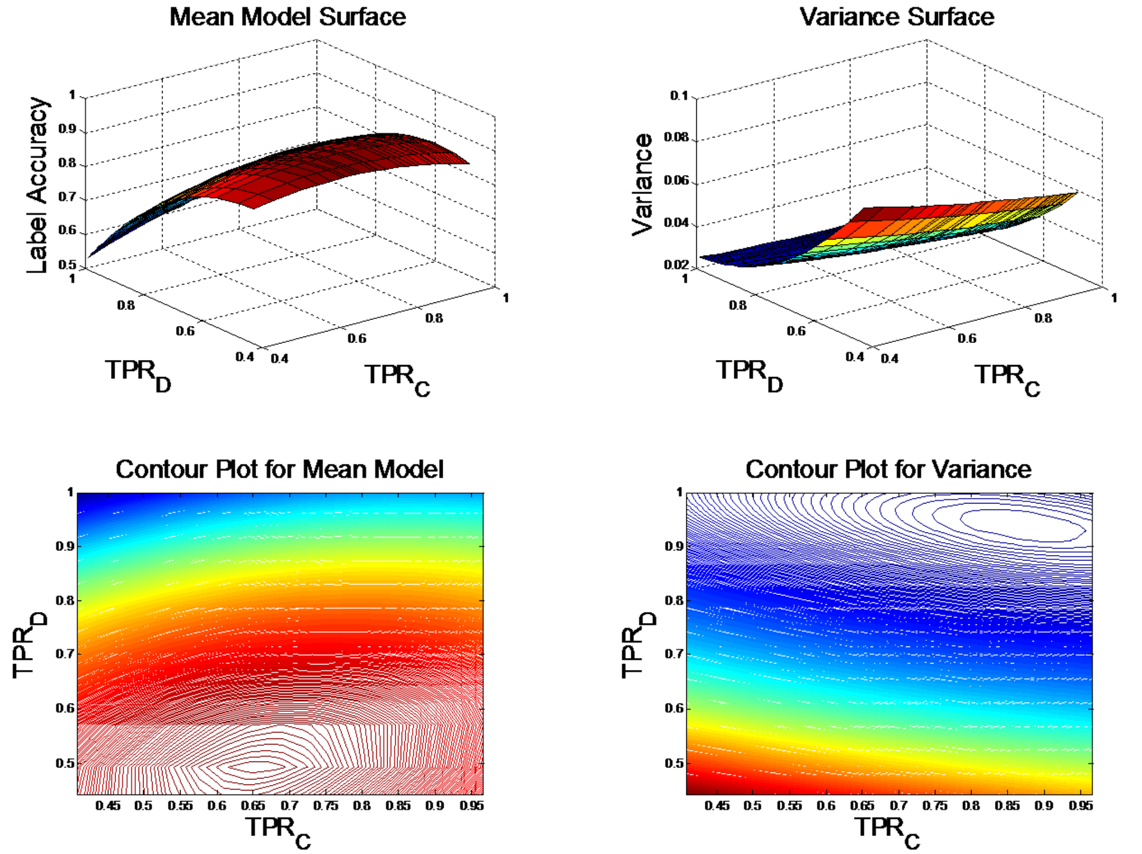


Figure 40: Surface, Contour Plots for Using the Label Accuracy of Enemy for 2nd ROC Set

As shown by Figure 40, the highest label accuracy happens when TPR_D is around 0.5 and TPR_C is around 0.65 and the lowest variance occurs at the northeast quadrant of the variance model. This output implies an inverse relation between label accuracy and variance, but we can see much improved mean and variance from 2nd ROC curve set.

ANNs Method

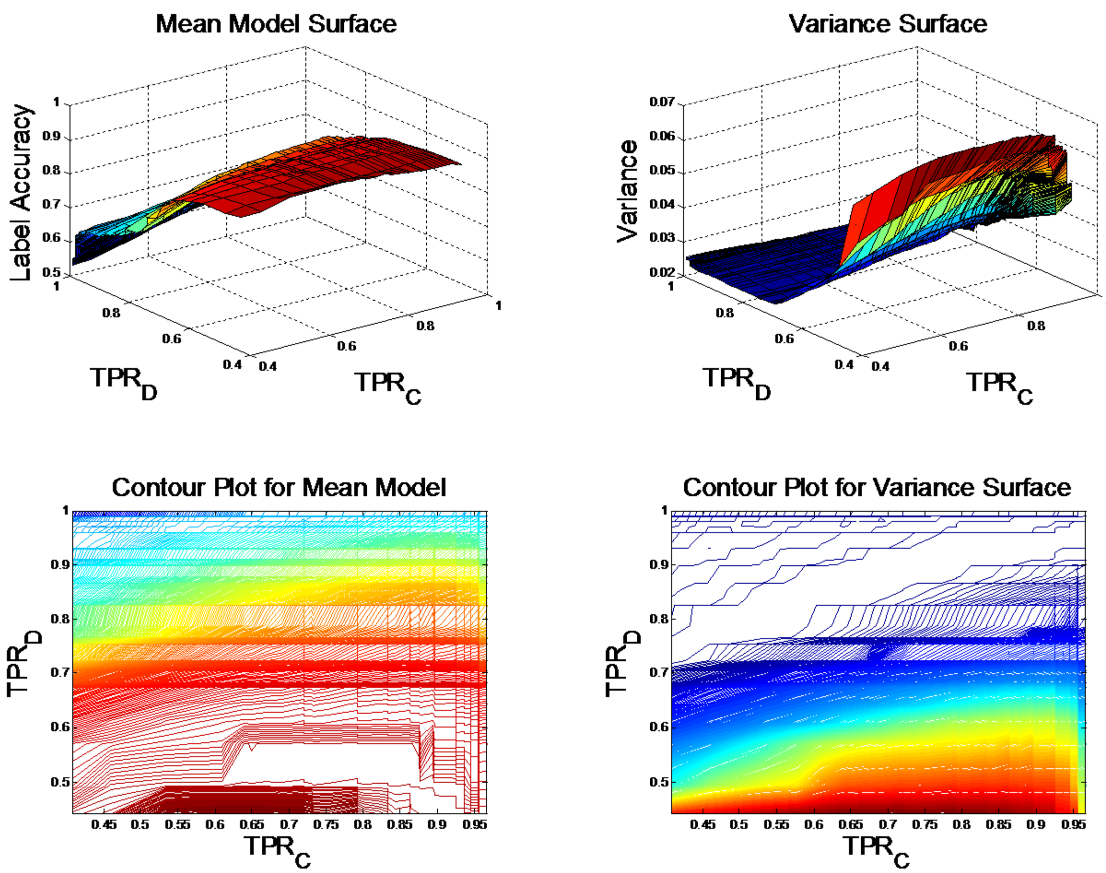


Figure 41: Surface, Contour Plots for Using the Label Accuracy of Enemy for 2nd ROC Set

Figure 41 shows more complex expected value and variance. The highest label accuracy happens when TPR_C is between 0.45 and 0.6 and the lowest variance occurs at the northeast quadrant of the variance model. Like Kim's method, this output indicates an inverse relationship of label accuracy and variance. Also, we can expect more accurate output from ANN Method based on Figure 41.

Comparison between Kim's method and ANNs Method

In order to evaluate ANN method, this research compares residual plot between Kim's method and the ANN method. These plots show that the residuals of Kim's are wider than the ANN method, and residuals of the ANN are scattered more constantly.

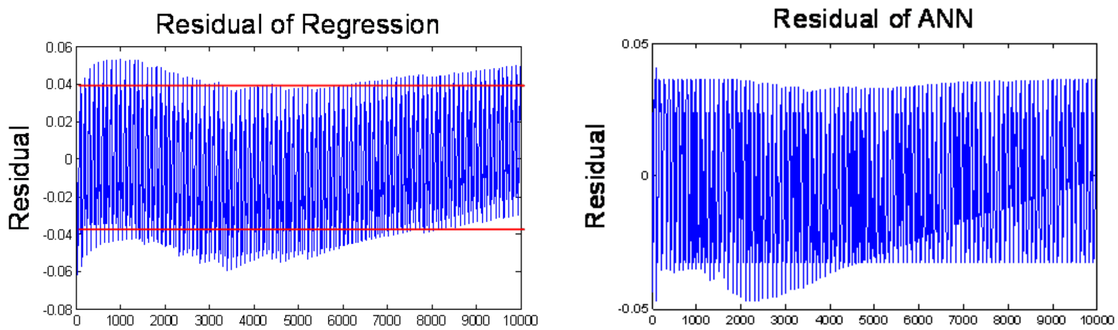


Figure 42: Residual plots of Kim's method and the ANN method (Note scales)

Optimal Points

Kim's method

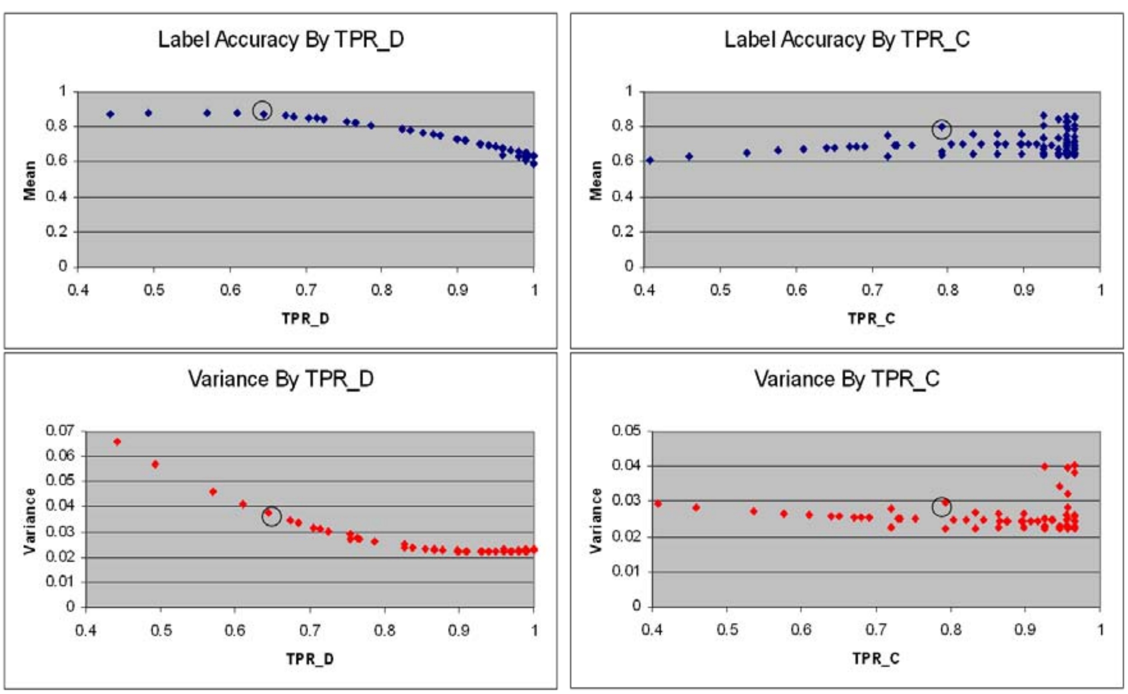


Figure 43: Average Mean and Variance by TPR_D and TPR_C (Kim's Method)

Figure 43 shows the highest label accuracies by TPR_D are distributed where TPR_D is between 0.4 and 0.8. However, the highest label accuracy has a high variance. As shown by label accuracy by TPR_D, the highest label accuracies by TPR_C are distributed at high variance. Thus, we should determine the point which has a high mean and appropriate variance. The optimal point takes place at the black circle where TPR_D is 0.644 and TPR_C is 0.7921. The plot of mean by variance in Figure 43 gives the same optimal point.

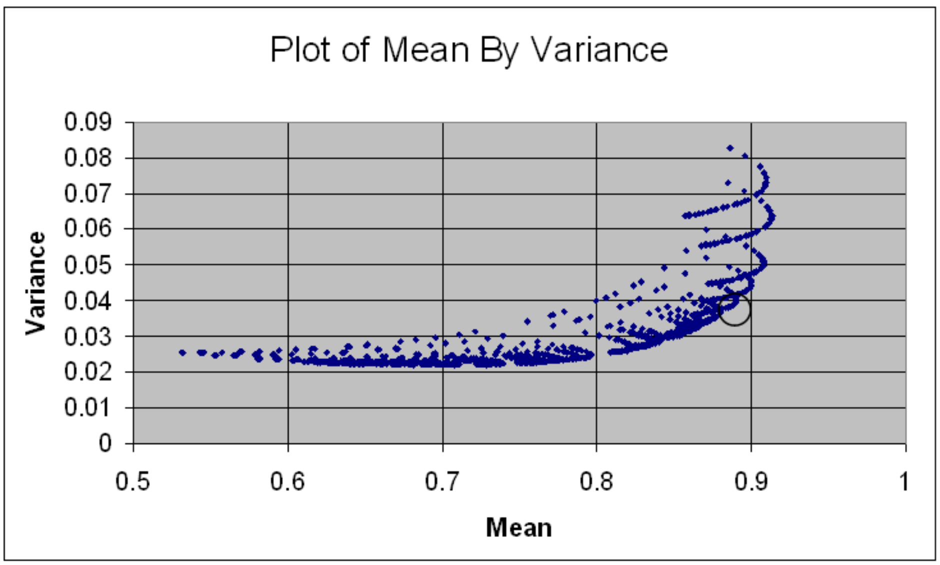


Figure 43: Plot of Mean by Variance

ANNs method

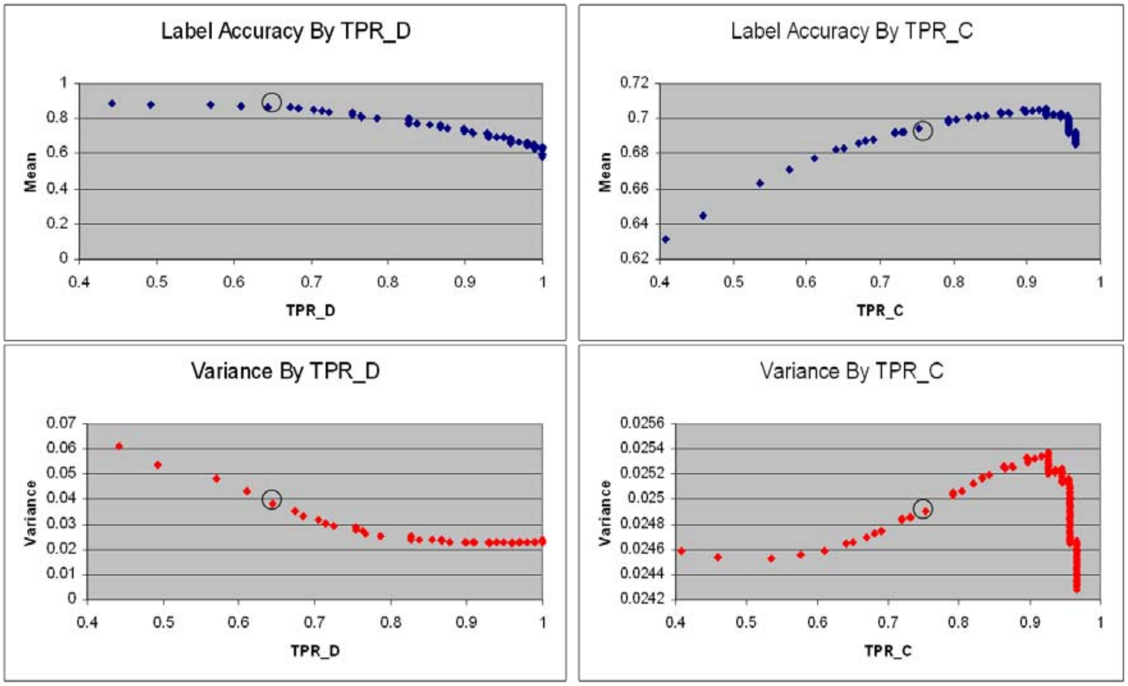


Figure 45: Average Mean and Variance by TPR_D and TPR_C (ANNs Method)

Like Kim's method, the highest label accuracies by TPR_D occurs where TPR_D is between 0.4 and 0.8. However, the highest label accuracy has a high variance. As shown by label accuracy by TPR_D, the highest label accuracies by TPR_C are distributed at the high variance. Thus, we should determine the point which has a high mean and appropriate variance. The optimal point takes place at the black circle where TPR_D is 0.644 and TPR_C is 0.7525. The plot of mean by variance in Figure 46 shows the same optimal point.

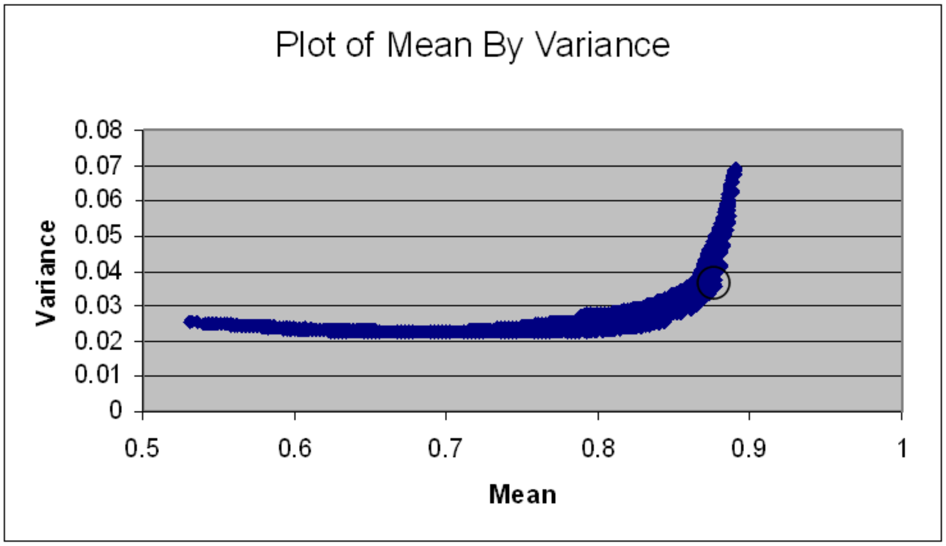


Figure 46: Plot of Mean by Variance

The solutions of Kim's method are the 1505th, 1605th and 1705th combinations, and the ANNs method is the 416th combination. Table12 shows Kim's solutions have a higher mean value, also have a higher variance.

Table 12: Solution of both Method

Method	Comb#	TPR_D	TPR_C	FPR_D	FPR_C	Mean	Variance
	1505	0.644	0.7921	0.0025	0.008	0.888276	0.038902
Kim	1605	0.644	0.7921	0.0025	0.0085	0.888276	0.038902
	1705	0.644	0.7921	0.0025	0.009	0.888276	0.038902
ANN	416	0.644	0.7525	0.0025	0.0075	0.875234	0.038297

Label accuracy of Friendly ($P(F|"F")$ ***)***

Kim's Method

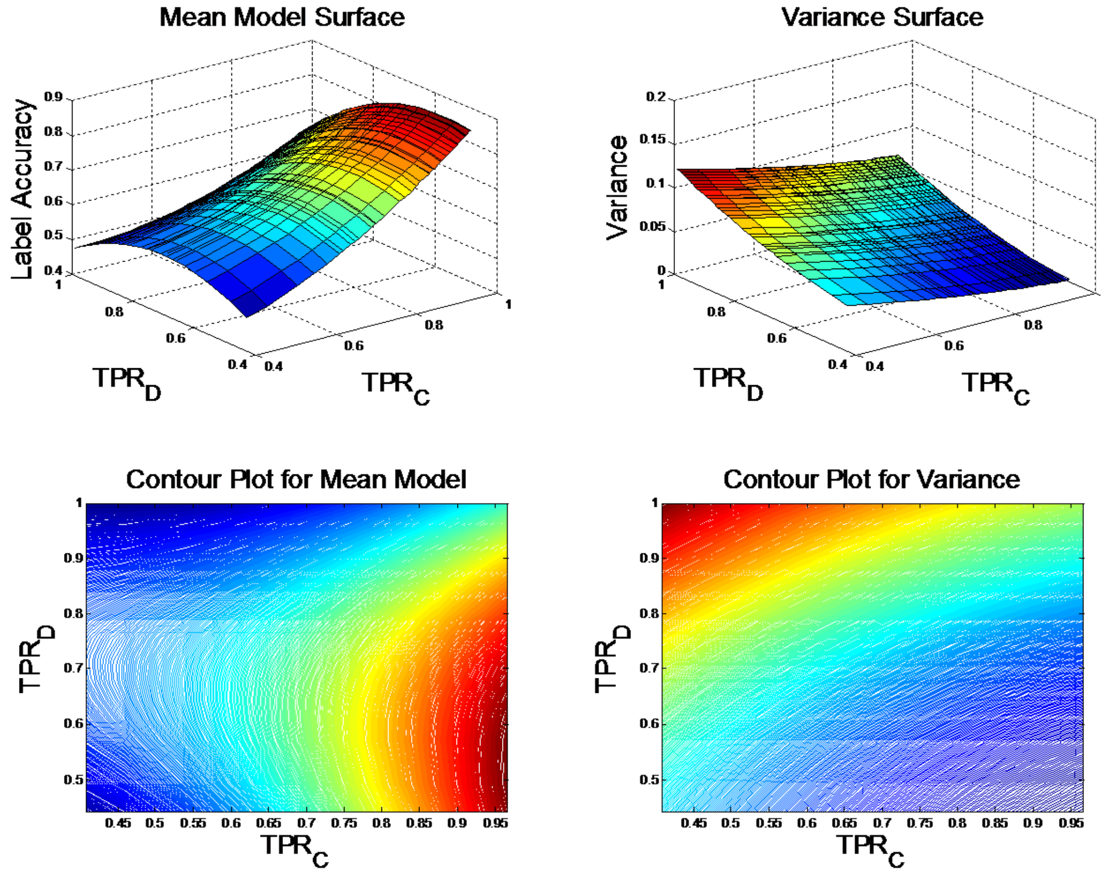


Figure 47: Surface, Contour Plots for Using the Label Accuracy of Friendly for 2nd ROC Set

As shown by Figure 47, the highest label accuracy happens when TPR_D is around 0.6 and TPR_C is around 1.0 and the lowest variance turns out at the southeast quadrant of the variance model. There appears to be a negative relationship between label accuracy and variance. Thus, we can find optimal point more easily than previous label accuracy for enemy.

ANNs Method

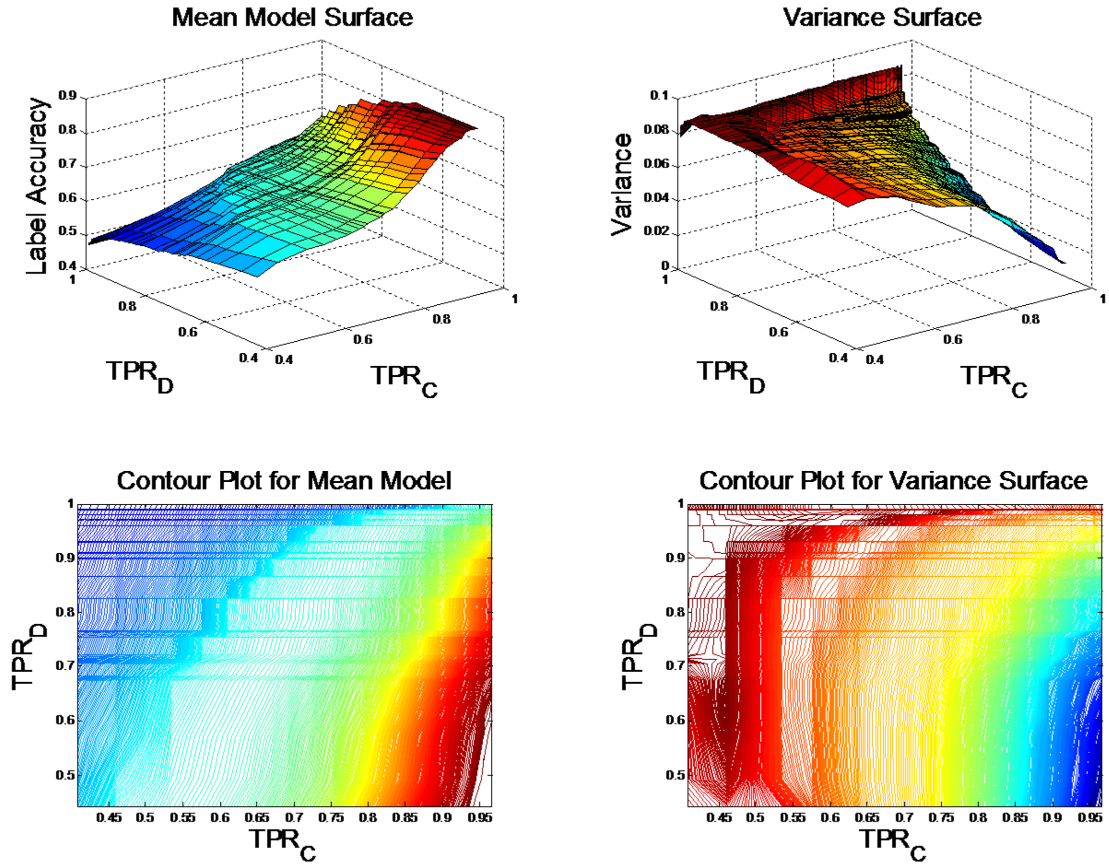


Figure 48: Surface, Contour Plots for Using the Label Accuracy of Friendly for 2nd ROC Set

Like plots of label accuracy for enemy, the above plots are more complex than plots of Kim's method. Figure 48 shows that the highest label accuracy occurs when TPR_D is around 0.5, and TPR_C is between 0.95 and 1.0. The lowest variance happens when TPR_D is between 0.5 and 0.6, and TPR_C is between 0.95 and 1.0. We need better expected value and lower variance. Thus, we can say that TPR_D between 0.95 and 1.0 and TPR_C between 0.4 and 0.55 are good point for label accuracy of friend. Additionally, we can expect more accurate output from ANN, based on Figure 48.

Comparison between Kim's method and ANNs Method

In order to evaluate ANNs method, this research compares again residual plot between Kim's method and ANNs method. These plots show that the residuals of Kim's are wider than ANNs method and residuals are much greater for friendly accuracy than for enemy label accuracy.

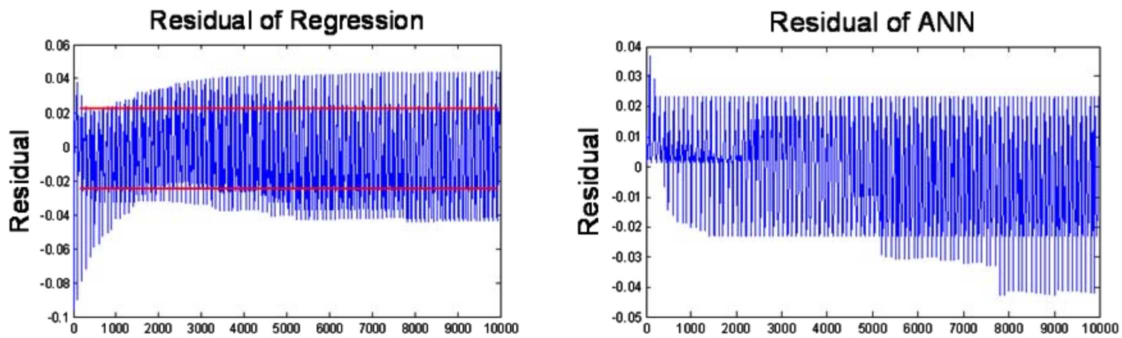


Figure 49: Residual plots of Kim's method and ANNs method

Optimal Points

Kim's Method

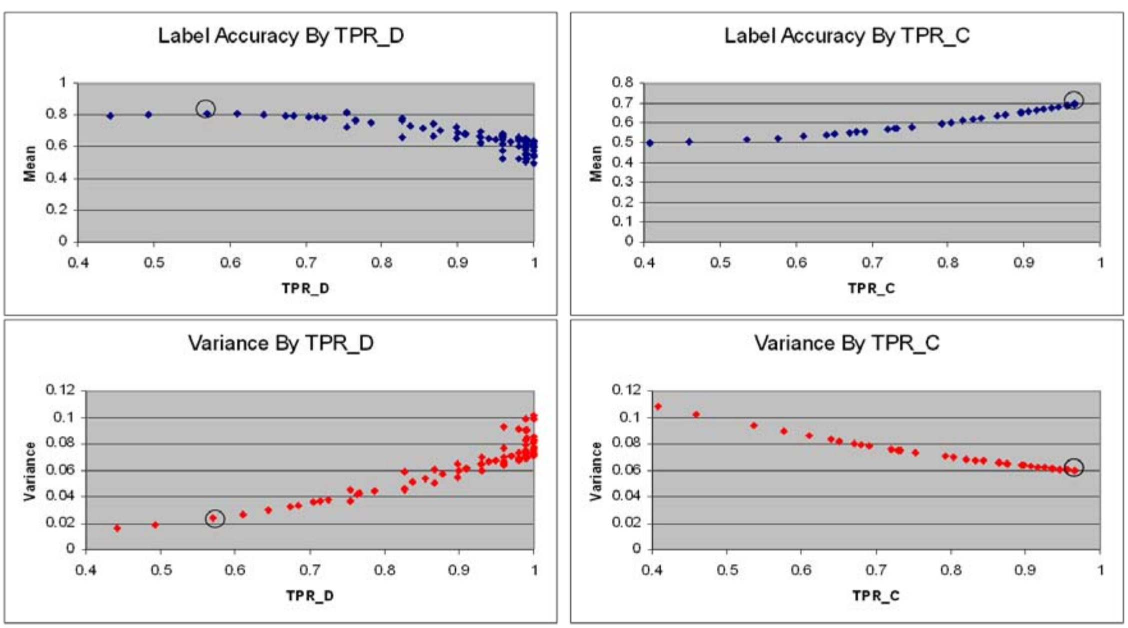


Figure 50: Average Mean and Variance by TPR_D and TPR_C (Kim's Method)

Figure 50 shows the higher label accuracies and the lower variance occurs where TPR_D is 0.5694 and TPR_C is 0.9667. Thus, the optimal point takes place at the black circle that TPR_D is 0.5694 and TPR_C is 0.9667. The plot of mean by variance in figure 51 shows the same optimal point.

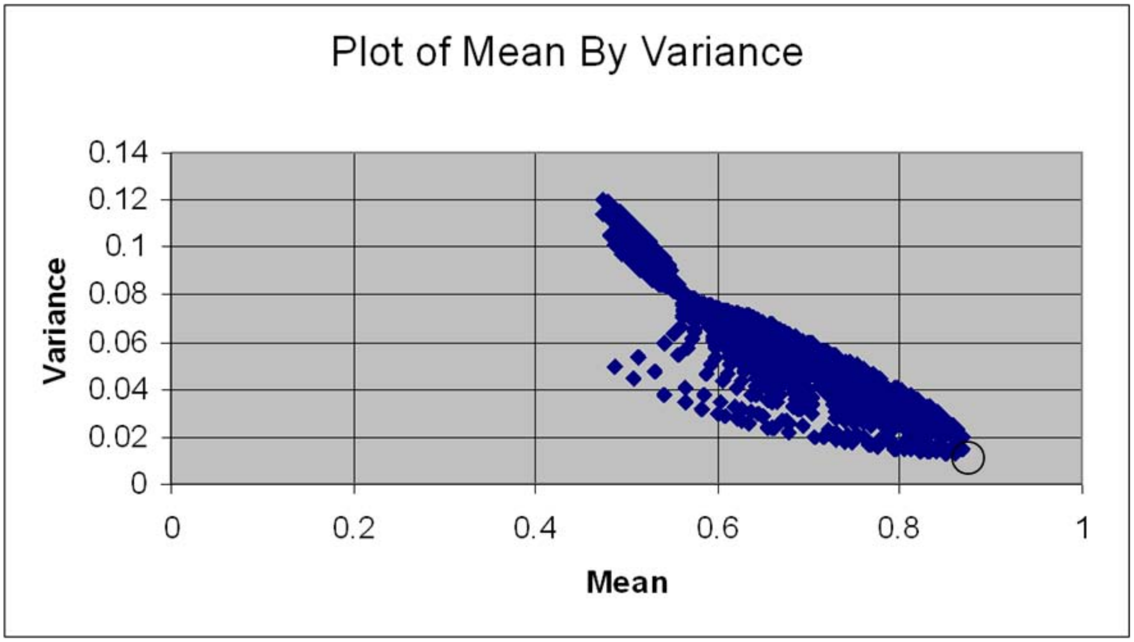


Figure 51: Plot of Mean by Variance

ANNs Method

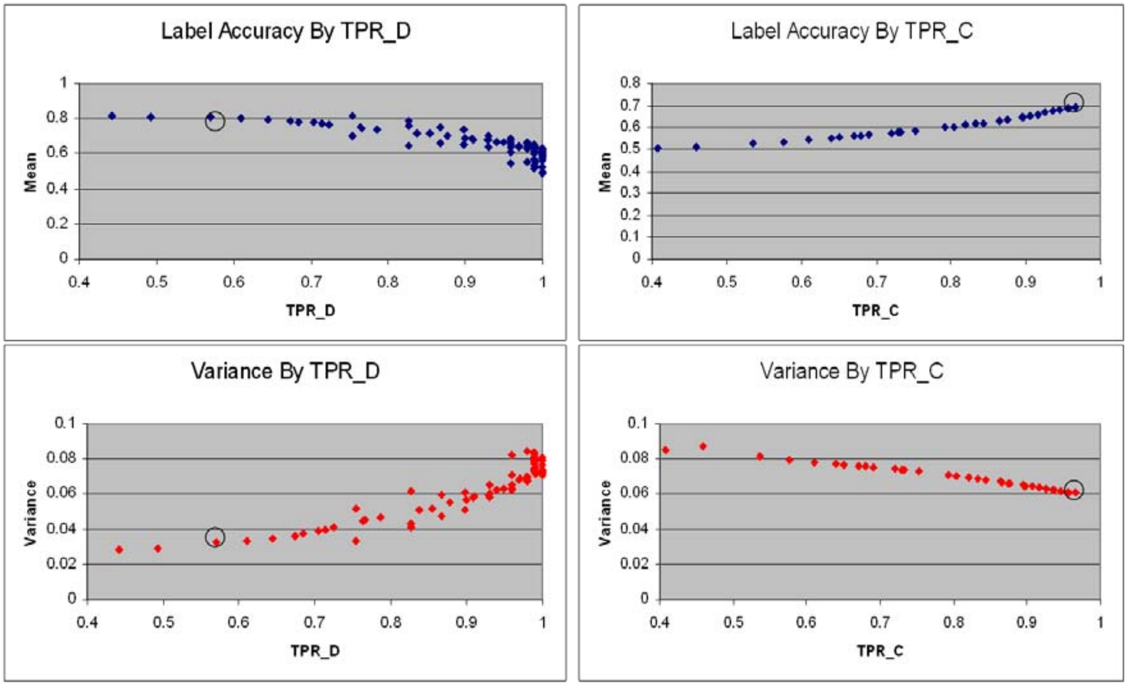


Figure 52: Average Mean and Variance by TPR_D and TPR_C (ANNs Method)

Like Kim's method, the highest label accuracies and the lower variance occur where TPR_D is 0.5694 and TPR_C is 0.9667. Thus, the optimal point takes place at the black circle that TPR_D is 0.5694 and TPR_C is 0.9667. The plot of mean by variance in Figure 53 shows the same optimal point.

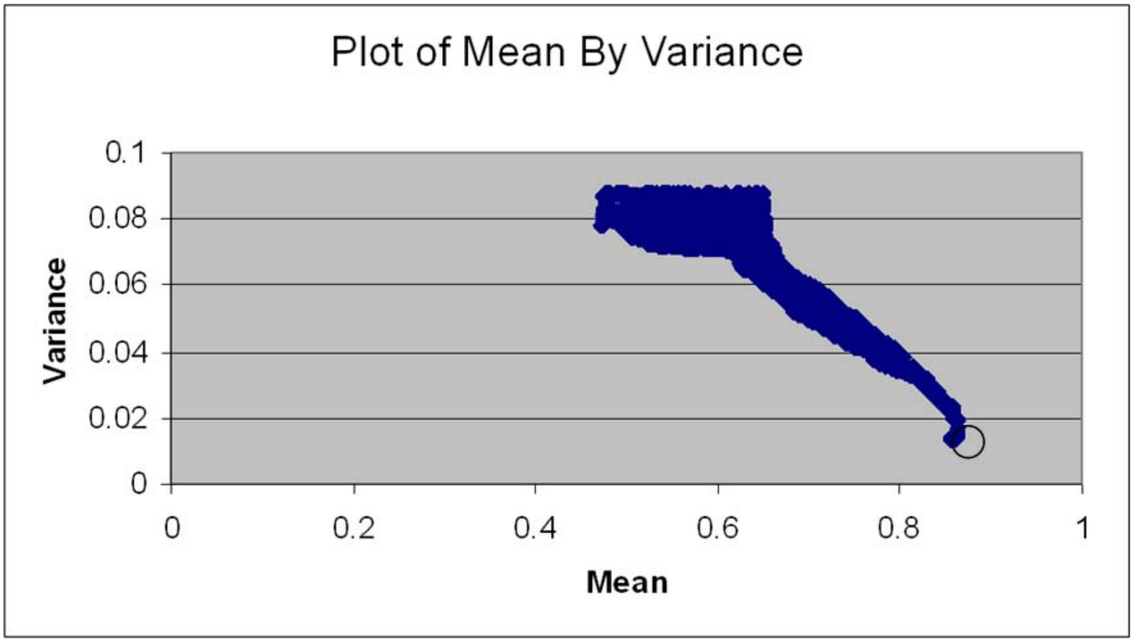


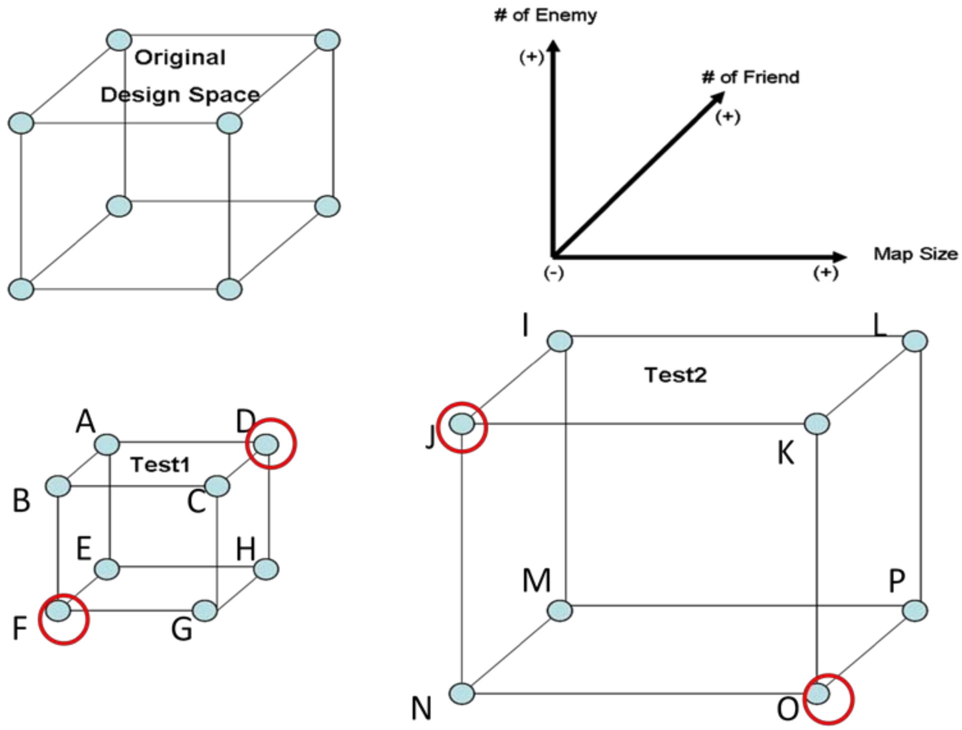
Figure 53: Plot of Mean by Variance

The solutions of Kim's method are from the 9103rd to the 9903rd combinations, and the ANNs method is the 9103rd combination. Table 13 shows Kim's solutions have a higher mean value, but also have a higher variance.

Table 13: Solution of both Method

Method	Comb#	TPR_D	TPR_C	FPR_D	FPR_C	Mean	Variance
	9103	0.5694	0.9667	0.0015	0.046	0.868316	0.019595
	9203	0.5694	0.9667	0.0015	0.0465	0.868316	0.019595
	9303	0.5694	0.9667	0.0015	0.047	0.868316	0.019595
	9403	0.5694	0.9667	0.0015	0.0475	0.868316	0.019595
Kim	9503	0.5694	0.9667	0.0015	0.048	0.868316	0.019595
	9603	0.5694	0.9667	0.0015	0.0485	0.868316	0.019595
	9703	0.5694	0.9667	0.0015	0.049	0.868316	0.019595
	9803	0.5694	0.9667	0.0015	0.0495	0.868316	0.019595
	9903	0.5694	0.9667	0.0015	0.05	0.868316	0.019595
ANN	9103	0.5694	0.9667	0.0015	0.046	0.863001	0.019356

Confirmation Experiments



	Map size		# of Enemy		# of Friendly	
	-	+	-	+	-	+
Original	100	1000	5	40	5	40
Test1	10	100	1	3	1	3
Test2	5000	10000	80	500	80	500

Figure 54: Notional Example of Design Space and the Table of Confirmation Experiments

The second part of the experiment did not suggest an obviously better model between Kim's and the ANN. Thus, we need an expanded experiment in order to determine the better model. The confirmation experiments, with regards to the 1st and 2nd ROC sets, are performed in different ROI surroundings: (1) a smaller Design space (Test1) which has small map size, number of enemies and number of friendly, and (2) a larger Design space (Test2) which has big map size, number of enemies and number of friendly, that is, 'Test1' is a inner design space of original and 'Test2' is a outer design space of the original problem. This confirmation experiment for two MoPs is conducted together and values for both methods are also reported together. The confirmation experiments are performed at two points of the inner spaces (D and F) and two points of the outer spaces (J and O).

Table 14: Test Points of Confirmation Experiments

Test point	Map size	# of Enemy	# of Friendly
D	100	3	3
F	10	1	1
J	5000	50	80
O	10000	80	80

Confirmation Experiments results of 1st ROC curve Set

Table 15: Output results of 1st ROC curve set

Response Type	Model	Comb#	Label Accuracy				Ave_Accuracy
			D	F	J	O	
Label Accuracy for Enemy	Kim's	703	0.4086	0.686	0.7175	0.2715	0.5209
	ANN	1303	0.4624	0.7073	0.764	0.3227	0.5641
Label Accuracy for Friendly	Kim's	603	0.6971	0.596	0.1853	0.194	0.4181
	ANN	1503	0.8647	0.7227	0.2796	0.191	0.5145

The blue shaded values are the best performance values (The higher label accuracy), when we do the confirmation of experiment with two methods (Kim’s and ANN) for a given design space. In most cases, the ANN method shows better performance. For the case of label accuracy for enemy, the ANN shows the higher label accuracies for all test points, additionally, label accuracies for friendly are also higher except for one case. Thus, the optimal points from ANN are more effective and reasonable to the decision makers, though it showed some bad cases for predicting plot, the ANN would be a better model for the 1st ROC curve set.

Confirmation Experiments results of 2nd ROC curve Set

Table 16: Output results of 2nd ROC curve set

Response Type	Model	Comb#	Label Accuracy				Ave_Accuracy
			D	F	J	O	
Label Accuracy for Enemy	Kim's	1505	0.9868	0.9712	0.9773	0.7625	0.9240
		1605	0.9862	0.9706	0.9772	0.7621	
		1705	0.9856	0.97	0.9771	0.7618	
	ANN	416	0.9868	0.9703	0.9762	0.7534	0.9217
Label Accuracy for Friendly	Kim's	9103	0.9494	0.8731	0.5048	0.3659	0.6730
		9203	0.9494	0.873	0.5047	0.3658	
		9303	0.9493	0.873	0.5045	0.3656	
		9403	0.9493	0.8729	0.5044	0.3655	
		9503	0.9493	0.8729	0.5043	0.3654	
		9603	0.9493	0.8729	0.5041	0.3653	
		9703	0.9492	0.8727	0.504	0.3651	
		9803	0.9492	0.8727	0.5039	0.365	
		9903	0.9492	0.8726	0.5037	0.3649	
	ANN	9103	0.9494	0.8731	0.5048	0.3659	0.6733

The blue shaded values represent again the best performance values when we do the confirmation of experiment with two methods for a given design space. In the most cases, the ANN showed the higher label accuracies, however, for three cases the ANN method showed lower label accuracies for enemy. Even if Kim's method has higher label accuracy for enemy, the differences between Kim's and the ANN are very small. Thus, 2^{nd} ROC curve set also suggests the ANN method is the better model.

Summary of experiment results

In this chapter, the experiments were taken in two parts. The first part validated Kim's method using analytic method. The second part was carried out using two different ROC curve sets with the three MoPs and two different methods as explained in previous chapters. The summary of experiments and results follow;

- The output analysis shows no difference between simulation and analytic methods, thus, we can conclude Kim's model is valid. Though Kim's simulation model is brilliant, its logic is complex and takes too much time (MATLAB running time increases significantly with map size), whereas the analytic method with ANNs is simple, accurate, and quick regardless of map size.
- In the case of label accuracy for enemy, the optimal solutions of Kim's method gave us the higher expected value and the higher variance. In addition, the residuals of Kim's were distributed more widely.
- In the case of label accuracy for friendly, each ROC curve set showed a different solution. 1^{st} ROC set gave us a higher expected value and a lower variance for Kim's method. 2^{nd} ROC set gave us a lower expected value and also the lower variance a ANNs method.

- In case of label accuracy for clutter, each ROC curve set showed very high label accuracy and low variances.
- Based on confirmation experiments, we can say the ANN model works well at 1st ROC curve set which is a normal ROC curve for classification and a little better one for detection, and ANNs model works well again at 2nd ROC curve set which is much improved ROC curves for both but still the detection curve is better than the classification's curve.

All results show that the expected value of optimal threshold combination is higher in the Kim's method. However, the unexplained variance is also higher as shown in residual plots. Thus, if we only try to consider mean and variance model with the controllable variables, then Kim's model could be a better model for the 1st ROC curve set. However, output result of 2nd ROC set indicates that ANNs is the better model, since its variance is smaller, moreover, the confirmation experiments show that the optimal solutions came from the ANN are more effective and reasonable to the decision makers. This is because 1st ROC curve is more close to real battlefield. As a result, we can conclude the ANN method has the better performance for CID modeling.

V. Summary and Conclusions

Many studies related to CID have the same goal: to maximize combat/mission effectiveness while reducing total casualties due to enemy action and collateral damage [4]. The objectives of this research were: (1) validation of Kim's simulation method applying an analytic method and (2) comparing the two models with three measures of performance (label accuracy for enemy, friendly, and clutter). Considering the features of CID, input variables were defined as two controllable (threshold combination of detector and classifier) and three uncontrollable (map size, number of enemies and friendly).

For CID modeling this research employed the following assumptions: (1) each detector and classifier occupies a predetermined ROC curve, (2) a neutral force and civilian are in the clutter, (3) there are three characteristics in a virtual ROI such as: enemy object, a friendly object, and clutter, (4) all entities have to be declared one of these and no entity can be non-declared [4].

The first set of experiments considers Kim's method using an analytical method. In order to create response variables, Kim's method uses Monte Carlo simulation. The output results showed no difference between simulation and the theoretical method. Kim's simulation logic is complex and takes too much time, whereas the analytic method is simple, accurate and quick regardless of design space size. Thus, we can say simulation method is not necessary if analytic solution is possible, although Kim's model is valid.

The second set of experiments compared the measures of performance (Label accuracy for enemy, friendly and clutter) between Kim's and ANNs method. To find optimal combinations of threshold, Kim's model uses regression with a combined array design, whereas the ANNs method uses ANN with a crossed array design. In the case of label accuracy for enemy, Kim's solution showed the higher expected value, however it also showed a higher variance. Additionally, the differences between actual plot and predicted plot were high for Kim's model. This leads to an unexplained variance.

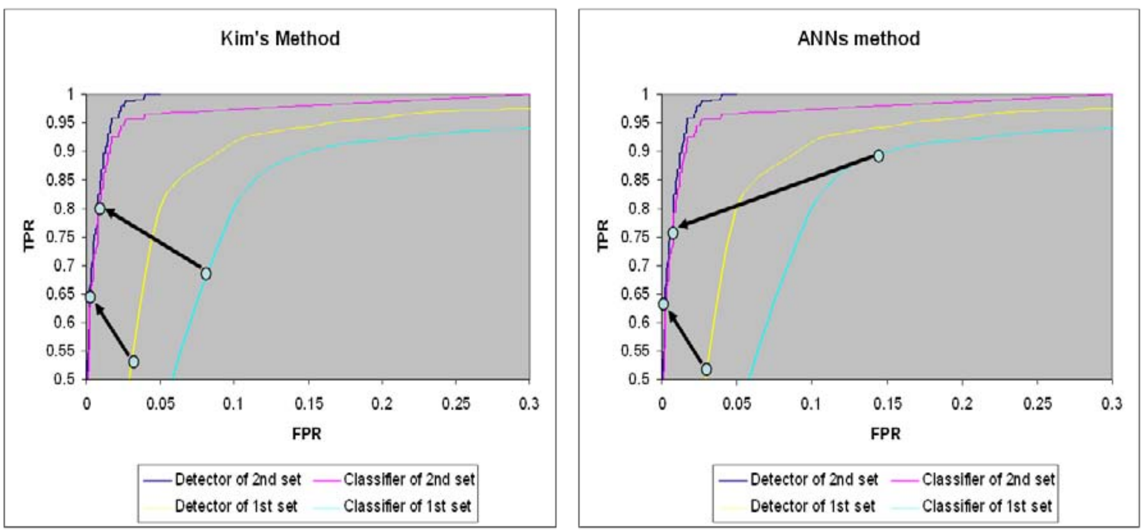


Figure 55: The Movement of the optimal points for Each Techniques (Label Accuracy for Enemy)

The optimal points for Kim's detector and classifier in Figure 55 moved to the points which allow higher TPR with lower FPR (northwest direction), however, the optimal points for ANNs method moved to a point which has lower TPR with lower FPR. For the detector, the optimal points occur where TPR_D is between 0.5 and 0.65, since the higher TPR_D also has the higher FPR. For the classifier, the optimal points did not occur at the highest TPR_C. The expected values of label accuracy for enemy are always higher for Kim's model, but the variances are also higher. Thus, in the case of enemy label accuracy, if the decision maker prefers a higher expected value, then Kim's model would be a better model, however, if the decision maker prefers the lower variance, ANNs model would be a better model.

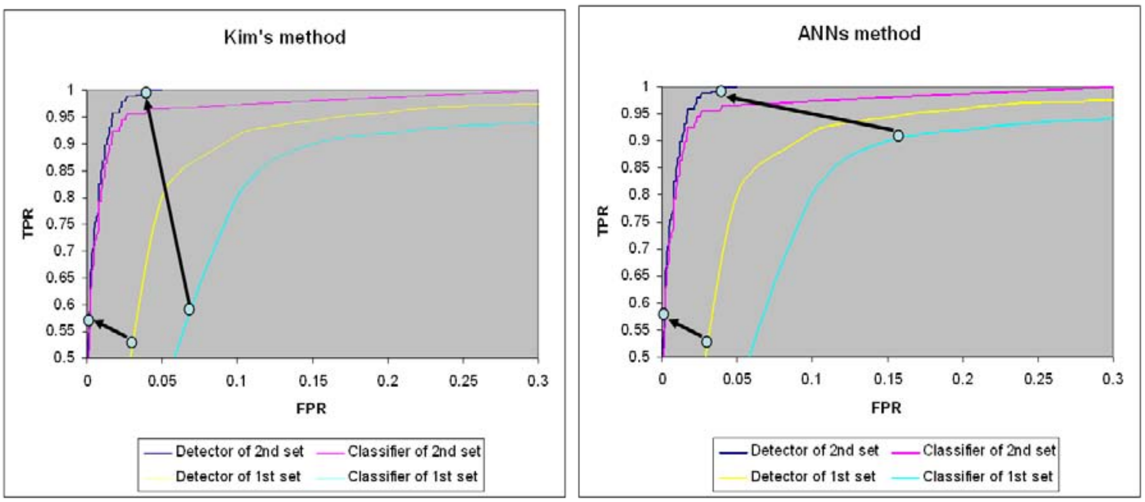


Figure 56: The Movement of the optimal points for Each Techniques (Label Accuracy for Friendly)

The optimal points for both Kim's and ANNs detector and classifier in Figure 56 moved to the points which allow higher TPR with lower FPR (northwest direction). For the detector, the optimal points occur where TPR_D is between 0.5 and 0.6, though a higher TPR_D has a lower FPR compared with 1st ROC curve set. For the classifier, the optimal points occur at the highest TPR_C regardless of models. 1st ROC set gave a higher expected value and a lower variance for Kim's model, and 2nd ROC set gave a lower expected value with small difference and the lower variance for ANNs model. Thus, in the case of friendly label accuracy, Kim's model would be a better model for the normal ROC curve set, however, ANNs model would be a better model for the improved ROC curve set.

Confirmation of experiments suggests a more detailed evaluation for both models. Based on Table 15 and 16 of Chapter 4, the ANN model showed a better performance for the 1st ROC set, and the 2nd ROC set. Thus, the ANN method would be the better model compared with Kim's model, since confirmation experiments show that the optimal solutions came from the ANN are more effective and reasonable to the decision makers. As a result, we can conclude the ANN method performs better in CID modeling.

In conclusion, if an analytic solution is possible then simulation is not necessary. The evaluation of a CID model could be changed by setting of design space and preference of decision maker. This is because a CID model of higher expected value does not guarantee a lower variance and measures of performance on CID vary by circumstances of the battlefield.

For further research, we can apply a new model for CID, since this research only considered one new method for modeling. Though this paper simplifies Kim's simulation using an analytic method and suggests a new prediction model for CID, the area for CID research is still ripe for experimentation, since we can apply a multitude of different factors (signal and decision factors) in the ROC curve [4].

APPENDIX A: MATLAB® CODE

A. Analytic model

% This Thesis Code is made by the author.

```
function[TagforReg,Tag, cvector,dvector,evector] = Analytic()


howmany = 1;
% % %          % %threshold = [.1 .2 .3 .4 .5 .6 .7 .8 .9 1; 0.8 0.92 0.94 0.95 0.96 0.97 0.98 0.99 0.995 1; .1
.2 .3 .4 .5 .6 .7 .8 .9 1; 0.3 0.52 0.7 0.8 0.85 0.89 0.92 0.95 0.97 1 ];
%          threshold = [.1 .2 .3 .4 .5 .6 .7 .8 .9 1; 0.8 0.92 0.94 0.95 0.96 0.97 0.98 0.99 0.995 1; .1 .2 .3 .4 .5
.6 .7 .8 .9 1; 0.8 0.92 0.94 0.95 0.96 0.97 0.98 0.99 0.995 1 ];
%
%          a = [0 threshold(1,:)]';
%          b = [0 threshold(2,:)]';
%          c   = a;
%          d = [0 threshold(4,:)]';
%          A=[]; B=[]; C=[]; D=[];
%
%          for i = 1:10
%             for j = 1:10
%                aa(j,i)=a(i) + ((a(i+1)-a(i))/10)*j;
%                bb(j,i)=b(i) + ((b(i+1)-b(i))/10)*j;
%                dd(j,i)=d(i) + ((d(i+1)-d(i))/10)*j;
%             end
%             A = [A;aa(:,i)];
%             B = [B;bb(:,i)];
%             D = [D;dd(:,i)];
%          end
%          C=A;
%          threshold_d = [A,B];
%          threshold_c = [C,D];

                    load 'new_threshold.mat' threshold_d;
                    load 'new_threshold.mat' threshold_c;

D = fullfact([100 100 2 2 2 2]);

avector = threshold_d(:,2); %TPR for Dec
bvector = threshold_c(:,2); %TPR for Class
cvector = [100 1000]'; %Map size
%=====================================================================
dvector = [5 40]'; %number of enemy
evector = [5 40]'; %number of friend

fvector = threshold_d(:,1); %FPR for Dec
gvector = threshold_c(:,1); %FPR for Class
F = D(:,1);
G = D(:,2);

%           F = D(:,1)/size(threshold_d,1);
```

```
%             G = D(:,2)/size(threshold_c,1);
%

D = [D,F,G];


for i=1:size(D,1) %sets with test values
    D(i,1)=avector(D(i,1),1);
    D(i,2)=bvector(D(i,2),1);
    D(i,3)=cvector(D(i,3),1);
    D(i,4)=dvector(D(i,4),1);
    D(i,5)=evector(D(i,5),1);
    D(i,7)=fvector(D(i,7),1);
    D(i,8)=gvector(D(i,8),1);

end

AnalyticY= [];
for i =1:size(D,1)
 % Label accuracy for Enemy
 Acc1(i) =
((D(i,1)*D(i,2)*D(i,4)/D(i,3))/(D(i,1)*D(i,2)*D(i,4)/D(i,3)+D(i,1)*D(i,8)*D(i,5)/D(i,3)+D(i,7)*0.5*(D(i,3)
-D(i,4)-D(i,5))/D(i,3)))';

% Label accuracy for Friendly
Acc2(i) = ((D(i,1)*(1-D(i,8))*D(i,5)/D(i,3)))/((D(i,1)*(1-D(i,8))*D(i,5)/D(i,3))+(D(i,1)*(1-
D(i,2))*D(i,4)/D(i,3))+(D(i,7)*0.5*(D(i,3)-D(i,4)-D(i,5))/D(i,3)))';

% Label accuracy for Clutter
Acc3(i) = ((1-D(i,7))*(D(i,3)-D(i,4)-D(i,5))/D(i,3))/((1-D(i,7))*(D(i,3)-D(i,4)-D(i,5))/D(i,3)+(1-
D(i,1))*0.5*D(i,4)/D(i,3)+(1-D(i,1))*0.5*D(i,5)/D(i,3))';
end
save('Acc1','Acc1');
save('Acc2','Acc2');
save('Acc3','Acc3');
```

B. Regression

% This Thesis Code is made by Kim (2007). And the author used it for this research

```
%inputs are A, Response, and Vnames                    <----------user input
%it doesnn't matter if A has leading ones

clc;
clear Bhat Yhat e SSres MSres SSreg MSreg SSt Fo Fstat alpha C H X r d;
clear ePRESS Si2 Rstud t nvector groupnum Ybarvector SSpe ANOVA Xhatp;
clear Yhatp U Z xi xerror yerror Tcrit BoxCoxusedlamda BoxCoxusedlog;
clear leveragepoints Cooks DFFITS Cooksinfluence DFFITSinfluence;
clear DFBETASinfluence DFBETAS DFBETAcountries V R Z Rstud ePRESS;
clear Yhata PRESS;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Switches
   GRAPHS=1;% 0 is off                                  <----------user input
   BOXCOX=0;% 0 is off                                   <----------user input
   ALLREG=0;% 0 is off                                  <----------user input
   LofFit=0;% 0 is off                              <----------user input
   Warnng=0;% 0 is off                                <----------user input
   GENLSQ=0;% 0 is off                                  <----------user input
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%add a column of ones to A if it needs one and get sizes of A (n by p)
   Y=Response;
   n=size(A,1);
   if A(:,1)~=ones(n,1)
      A=[ones(n,1) A];
   end
   p=size(A,2);
   globalp=p;
   Filter = int8(ones(1,p));

   %Filter out certain regressors - uncomment to "eliminate"
%     Filter(1,1)=0;% filter B0                         <----------user input*
%     Filter(1,2)=0;% filter B1                         <----------user input
%     Filter(1,3)=0;% filter B2                         <----------user input*
%     Filter(1,4)=0;% filter B3                         <----------user input*
%     Filter(1,5)=0;% filter B4                         <----------user input
%     Filter(1,6)=0;% filter B5                         <----------user input*
%     Filter(1,7)=0;% filter B6                         <----------user input
%     Filter(1,8)=0;% filter B7                         <----------user input

   X=A;
   for i=p:-1:1
      if Filter(1,i)==0
         X(:,i) = [];
      end
   end
   p=size(X,2);

   explist=ones(1,p);
   Xform=int8(zeros(1,p));
   %Pick regressors to transform - uncomment to Xform via Box-Tidwell
%%%%%%%%%%%%Do not transform x0 via Box Tidwell
%         Xform(1,2)=1;% Xforms x1 via Box-Tidwell               <----------user input
%         Xform(1,3)=1;% Xforms x2 via Box-Tidwell               <----------user input
%         Xform(1,4)=1;% Xforms x3 via Box-Tidwell               <----------user input
%         Xform(1,5)=1;% Xforms x4 via Box-Tidwell               <----------user input
%         Xform(1,6)=1;% Xforms x5 via Box-Tidwell               <----------user input
%         Xform(1,7)=1;% Xforms x6 via Box-Tidwell               <----------user input
%         Xform(1,8)=1;% Xforms x7 via Box-Tidwell               <----------user input

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
if Warnng==0
   warning off;
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%
%General Least Squares
if GENLSQ==1
   Save=X;
   V=cov(X');
   invV=(V)^-1;
   Bhatz=((X'*invV*X)^-1)*X'*invV*Y;
   K=(V)^.5;%  <--------- if covariances are negative, sqrts will be imaginary.
   Bee=((K)^-1)*X;
   bigZ=Bee*Bhatz; % <-------------also imaginary

   SSresz=bigZ'*bigZ-Bhatz'*Bee'*bigZ;
   MSresz=SSresz/(n-p);

   SSregz=Bhatz'*Bee'*bigZ;
   MSregz=SSregz/(p-1);

   SStz=bigZ'*bigZ;

   %Calculate F statistic for model
   alpha=.90;
   Foz=MSregz/MSresz;
   Fstatz=finv(alpha,p-1,n-p);
   Fpvaluez=1-fcdf(Foz,p-1,n-p);

   %R-squared
   R2z=SSregz/SStz;
   R2adjz=1-(SSresz/(n-p))/(SStz/(n-1));

   %Build table (see pg 80 in book for explanation)
   glmANOVA=zeros(4,6);
   glmANOVA(1,1)=SSregz;  glmANOVA(1,2)=p-1;    glmANOVA(1,3)=MSregz;
glmANOVA(1,4)=Foz; glmANOVA(1,5)=Fpvaluez;
   glmANOVA(2,1)=SSresz;  glmANOVA(2,2)=n-p;    glmANOVA(2,3)=MSresz;
   glmANOVA(3,1)=SStz;    glmANOVA(3,2)=n-1;
   glmANOVA(4,1)=R2z;     glmANOVA(4,2)=R2adjz;


   clear  invV K Bee;
   X=Save;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%
%transformations on X -BoxTidwell
   alpha=.9;%                              <----------user input
   y=Y;

   leading=ones(n,1);
   for i=1:p

      if Xform(1,i)==1
         x=[leading, X(:,i)];
         px=size(x,2);
```

```
        a=1;
        olda=10;

        while abs(olda-a)>.00005
          %step 1
          bhat=((x'*x)\eye(px))*x'*y;
          yhat=x*bhat;
          C=(x'*x)\eye(px);
          SSres=y'*y-bhat'*x'*y;
          MSres=SSres/(n-px);
          To=abs(bhat(px,1)/sqrt(MSres*C(px,px)));
          Tcrit=tinv((alpha+(1-alpha)/2),n-px);

          %step 2
          w=x(:,px).*log(x(:,px));
          xw=[x,w];

          %step 3
          bhatw=((xw'*xw)\eye(px+1))*xw'*y;
          yhatw=xw*bhatw;

          %step 4
          Cx=(xw'*xw)\eye(px+1);
          SSresx=y'*y-bhatw'*xw'*y;
          MSresx=SSresx/(n-(px+1));

          Tox=abs(bhatw(px+1,1)/sqrt(MSresx*Cx(px+1,px+1)));
          Tcritx=tinv((alpha+(1-alpha)/2),n-(px+1));

          %step 5
          if To>Tcrit && Tox>Tcritx
            a=bhatw(px+1,1)/bhat(px,1)+a;
          else
            olda=a;
          end

          %step 6
          x(:,px)=x(:,px).^a;

        end
        explist(1,i)=a;
    end
end

for i=1:p
explist(1,i)=round(explist(1,i)*2)/2;

    if explist(1,i)>2
        explist(1,i)=2;
    end
    if explist(1,i)<(-2)
        explist(1,i)=(-2);
    end
end
```

```
  for i=1:p
    X(:,i)=X(:,i).^explist(1,i);
  end

clear x y olda To Tcrit Tox Tcritx w Cx bhatw;
clear MSresx SSresx MSres SSres yhatw bhat a xw yhat;
clear Xform leading %explist;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%
%transformations on Y -BoxCox

  if BOXCOX==1
    lamda=linspace(-2,2,21);
    lp=size(lamda,2);

    ydot=exp((1/n)*sum(log(Y)));

    for i=1:lp
      if lamda(1,i)~=0
        ytemp=(Y.^lamda(1,i)-1)./(lamda(1,i).*ydot^(lamda(1,i)-1));
      else
        ytemp=ydot.*log(Y);
      end
      bhat=((X'*X)\eye(p))*X'*ytemp;
      yhat=X*bhat;
      C=inv(X'*X);
      SSreslamda(1,i)=ytemp'*ytemp-bhat'*X'*ytemp;
    end

    lmin=min(SSreslamda);
    for i=1:lp
      if SSreslamda(1,i)==lmin
        location=i;
      end
    end
    if lmin~=0
      Y=(Y.^lamda(1,location)-1)/lamda(1,location);
      BoxCoxusedlamda=lamda(1,location)
    else
      Y=log(Y);
      BoxCoxusedlog=1
    end
    if GRAPHS==1
    figure(1)
    scatter(lamda,SSreslamda,'or', 'MarkerFaceColor','c');
    xlabel('Power Transformation Parameter Lamda');
    ylabel('SS_r_e_s'); title('SS_r_e_s vs. Lambda');
    end
  end
clear lp lmin ytemp location bhat yhat SSreslamda lamda ydot;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%
%fit model
```

```
  Bhat=((X'*X)\eye(p))*X'*Y;
  Yhat=X*Bhat;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%All possible regressions (p counts the intercept)
  if ALLREG==1
    clear All Nines Btemp mm nn U pall Bhata;

    AllReg=zeros(1,p);

    for i=1:p
      cmb=combntns(1:p,i);
      mm=size(cmb,1);
      nn=size(cmb,2);
      Btemp=zeros(mm,p);
      for j=1:mm
        for k=1:nn
          Btemp(j,cmb(j,k))=1;
        end
      end
      AllReg=[AllReg;Btemp];
    end

    clear mm nn;
    mm=size(AllReg,1);
    nn=size(AllReg,2);

    U=X; %U holds the original X
    for i=1:mm
      for j=nn:-1:1
        if AllReg(i,j)==0
          X(:,j) = [];
        end
      end

      pall=size(X,2);
      Bhata=((X'*X)\eye(pall))*X'*Y;
      Yhata=X*Bhata;
       e=Y-Yhata;
       H=X*((X'*X)\eye(pall))*X';
        for s=1:n
          ePRESS(s,1)=(e(s,1)/(1-H(s,s)))^2;
        end

      All(i,1)=Bhata'*X'*Y -(Y'*ones(n,1))^2/n;          %SSreg
      All(i,2)=Y'*Y-Bhata'*X'*Y;                          %SSres
      All(i,3)=All(i,1)+All(i,2);                         %SSt
      All(i,4)=All(i,1)/All(i,3);                         %R2
      All(i,5)=1-(All(i,2)/(n-pall))/(All(i,3)/(n-1));    %R2adj
      All(i,6)=sum(ePRESS);                               %PRESS

      X=U;
    end
    X=U;  %reset X
```

```
    numrgs=sum(AllReg')';
    tempM=ones(1,6);
    PandR2s=zeros(1,3);

    for i=1:p
      k=1;
      for j=1:mm
        if numrgs(j,1)==i
          tempM(k,:)=All(j,:);
          k=k+1;
        end
      end
        pickbiggest=max(tempM ,[] ,1);
        PandR2s(i,1)=i;            %the # of parameters used
        PandR2s(i,2)=pickbiggest(1,4); %R2
        PandR2s(i,3)=pickbiggest(1,5); %R2adj
    end

    if GRAPHS==1
    figure(2)
    plot(PandR2s(:,1),PandR2s(:,2),'r:o')
    hold on
    plot(PandR2s(:,1),PandR2s(:,3),'b:+')
    hold off
    xlabel('Number of Regression Coeficients');
    ylabel('R^2'); title('R^2 vs. Number of Regression Coefficients');
    legend('R^2','R^2 Adj.',2);
    end

    Nines=ones(mm,1)*9999999;
    All=[AllReg,Nines,All];
  else
    clear All;
  end
  clear nn mm nopt i j k Bhata Nines U pall cmb AllReg Btemp numrgs tempM;
  clear pickbiggest PandR2s;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
%perform ANOVA
  alpha=.95;%                       <----------user input

  C=(X'*X)\eye(p);

  SSres=Y'*Y-Bhat'*X'*Y;
  MSres=SSres/(n-p);

  SSreg=Bhat'*X'*Y-(Y'*ones(n,1))^2/n;
  MSreg=SSreg/(p-1);

  SSt=SSreg+SSres;

  %Calculate F statistic for model
  Fo=MSreg/MSres;
```

```
    Fstat=finv(alpha,p-1,n-p);
    Fpvalue=1-fcdf(Fo,p-1,n-p);

    %Perform marginal T test for each Bhat
    for i=1:p
        To(i,1)=Bhat(i,1)/sqrt(MSres*C(i,i));
        StdErr(i,1)=sqrt(MSres*C(i,i));
        Tcrit(i,1)=tinv((alpha+(1-alpha)/2),n-p);
        Tpvalue(i,1)=2*(1-tcdf(abs(To(i,1)),n-p));
    end

    %R-squared
    R2=SSreg/SSt;
    R2adj=1-(SSres/(n-p))/(SSt/(n-1));

    %Multicollinearity
%    Z=X;
%    Z(:,1)=[];

%    invR=corr(Z)\eye(p-1);
%     VIF=zeros(p,1);
%     for i=1:p-1
%         VIF(i+1,1)= invR(i,i);
%     end



    for i=1:p
        CIforBhat(i,1)=Bhat(i,1)-tinv((alpha+(1-alpha)/2),n-p)*sqrt(MSres*C(i,i));
        CIforBhat(i,2)=Bhat(i,1);
        CIforBhat(i,3)=Bhat(i,1)+tinv((alpha+(1-alpha)/2),n-p)*sqrt(MSres*C(i,i));
    end

    %Build table (see pg 80 in book for explanation)
    ANOVA=zeros(5+p,6);
    ANOVA(1,1)=SSreg;  ANOVA(1,2)=p-1;    ANOVA(1,3)=MSreg;  ANOVA(1,4)=Fo;
ANOVA(1,5)=Fpvalue;
    ANOVA(2,1)=SSres;  ANOVA(2,2)=n-p;    ANOVA(2,3)=MSres;
    ANOVA(3,1)=SSt;    ANOVA(3,2)=n-1;
    ANOVA(4,1)=R2;     ANOVA(4,2)=R2adj;
    for i=1:p
        ANOVA(5+i,1)=Bhat(i,1);
        ANOVA(5+i,2)=StdErr(i,1);
        ANOVA(5+i,3)=To(i,1);
        ANOVA(5+i,4)=Tcrit(i,1);
        ANOVA(5+i,5)=Tpvalue(i,1);
%        ANOVA(5+i,6)=VIF(i,1);
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
clear n p Filter Si2 SSres MSres SSreg MSreg SSt Fo Fstat ePRESS i r d t;
```

```
clear alpha disp residuals H Fpvalue C R2 R2adj dfssres dfsspe dfsslof;
clear nvector ttlvector Ybarvector m j N groupnum counter lofFo e;
clear lofFpvalue SSlof SSpe StdErr To Tstat Tpvalue Bhat Rstud I VIF;
clear invR Tcrit X LofFit ALLREG BOXCOX GRAPHS globalp Warnng jvector;
clear DFFITS Cooks GENLSQ Foz Fpvaluez SStz SSresz SSregz MSresz MSregz;
clear Yhata Bhata Fstatz R2z R2adjz Save s;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%
warning on;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%
```

C. Crossed Array Design

% This Thesis Code is made by Kim (2007). And the author used it for this research

```
%function [mean, variance, SN] = crossarray()

r = 2^4; % 3 noise factors with 2 levels
cross = zeros(size(Response,1)/r,r+4);


%==========================================  Make cross arry response
for i = 1: size(Response,1)/r
    for j = 1: r
        cross(i,j) = Response(i+10000*(j-1));
    end
    i
end


%==========================================  Make mean, variance, and S|N
for i = 1: size(Response,1)/r
    cross(i,r+1) = sum(cross(i,1:r))/r;
    cross(i,r+2) = var(cross(i,1:r));

    for j = 1: r
    y_sq(i,j) = 1 / cross(i,j)^2;
    y_sq2(i,j) = cross(i,j)^2;
    end


    cross(i,r+3) = -10*log10(1/r*(sum(y_sq(i,1:r))));
    cross(i,r+4) = 10*log10(1/r*(sum(y_sq2(i,1:r))));

    i
end

%===========Plotting===============================================
new_cross = [Tag(1:10000,1:2),cross(:,9:12)];

x3 = new_cross(1:100,1);
x4 = [];
for i = 1:100
```

```
    b = i*100-99;
    c = new_cross(b,2);
    x4 = [x4;c];
end
```

D. Artificial Neural Network

```
% This Thesis Code is made by author
%ANN for Thesis


T1 = [T1']; % Taget of Mean

P1 = cross(:,17);
P1 = [P1']; % Input Mean


MyNN1 = newff(minmax(P1),[hidden layer,1],{'logsig' 'logsig'});
MyNN1.trainParam.epochs =1000;
[MyNN1] = train(MyNN1,P1,T1);

 MyNN1.IW{:,:}
MyNN1.LW{:,:}
MyNN1.b{:,:}

 YTrained_Mean = sim(MyNN1,P1);

% Mean and variance model
for t = 1:size(x3,1)
    for r = 1:size(x4,1)
        z(t,r) = (YTrained_Mean(t+100*(r-1)))';
        v(t,r) = (YTrained_Variance(t+100*(r-1)))';
    end
end
figure(1)
surf(x4,x3,z)
title('Mean Model Surface','fontsize',20)
xlabel('TPR_C','fontsize',20)
ylabel('TPR_D','fontsize',20)
zlabel('Label Accuracy','fontsize',20)

figure(2)
surf(x4,x3,v)
title('Variance Surface','fontsize',20)
xlabel('TPR_C','fontsize',20)
ylabel('TPR_D','fontsize',20)
zlabel('Variance','fontsize',20)

figure(3)
contour(x4,x3,z,500)
title('Contour Plot for Mean Model','fontsize',20)
xlabel('TPR_C','fontsize',20)
ylabel('TPR_D','fontsize',20)
```

```
figure(4)
contour(x4,x3,v,500)
title('Contour Plot for Variance Surface','fontsize',20)
xlabel('TPR_C','fontsize',20)
ylabel('TPR_D','fontsize',20)
```

APPENDIX B: ROC THRESHOLD DATA FILE

SET1				SET2			
DETECTOR		CLASSFIER		DETECTOR		CLASSFIER	
FPR_D	TPR_D	FPR_C	TPR_C	FPR_D	TPR_D	FPR_C	TPR_C
0.01	0.1685	0.01	0.0803	0.0005	0.4422	0.0005	0.4082
0.02	0.3483	0.02	0.166	0.001	0.4932	0.001	0.4592
0.03	0.524	0.03	0.2542	0.0015	0.5694	0.0015	0.5354
0.04	0.6816	0.04	0.3431	0.002	0.6098	0.002	0.5758
0.05	0.8	0.05	0.4311	0.0025	0.644	0.0025	0.61
0.06	0.8443	0.06	0.5168	0.003	0.674	0.003	0.64
0.07	0.8656	0.07	0.5987	0.0035	0.684	0.0035	0.65
0.08	0.8825	0.08	0.6751	0.004	0.704	0.004	0.67
0.09	0.8996	0.09	0.7435	0.0045	0.714	0.0045	0.68
0.1	0.917	0.1	0.8	0.005	0.724	0.005	0.69
0.11	0.9268	0.11	0.8375	0.0055	0.754	0.0055	0.72
0.12	0.9321	0.12	0.8629	0.006	0.754	0.006	0.72
0.13	0.9362	0.13	0.8802	0.0065	0.764	0.0065	0.73
0.14	0.9403	0.14	0.8919	0.007	0.7667	0.007	0.7327
0.15	0.944	0.15	0.9	0.0075	0.7865	0.0075	0.7525
0.16	0.948	0.16	0.9067	0.008	0.8261	0.008	0.7921
0.17	0.9517	0.17	0.9118	0.0085	0.8261	0.0085	0.7921
0.18	0.9549	0.18	0.9153	0.009	0.8261	0.009	0.7921
0.19	0.9576	0.19	0.9178	0.0095	0.8379	0.0095	0.8039
0.2	0.96	0.2	0.92	0.01	0.854	0.01	0.82
0.21	0.9627	0.21	0.9231	0.0105	0.8673	0.0105	0.8333
0.22	0.9653	0.22	0.9263	0.011	0.8673	0.011	0.8333
0.23	0.9676	0.23	0.9292	0.0115	0.8771	0.0115	0.8431
0.24	0.9695	0.24	0.9317	0.012	0.8981	0.012	0.8641
0.25	0.971	0.25	0.9339	0.0125	0.8981	0.0125	0.8641
0.26	0.9722	0.26	0.9356	0.013	0.8994	0.013	0.8654
0.27	0.9731	0.27	0.937	0.0135	0.909	0.0135	0.875
0.28	0.9738	0.28	0.9381	0.014	0.9102	0.014	0.8762
0.29	0.9743	0.29	0.9391	0.0145	0.9292	0.0145	0.8952
0.3	0.975	0.3	0.94	0.015	0.9307	0.015	0.8967
0.31	0.9761	0.31	0.9412	0.0155	0.9308	0.0155	0.8972

0.32	0.9773	0.32	0.9425	0.016	0.9308	0.016	0.8972
0.33	0.9786	0.33	0.9437	0.0165	0.9401	0.0165	0.9065
0.34	0.9798	0.34	0.9448	0.017	0.9495	0.017	0.9159
0.35	0.9809	0.35	0.9459	0.0175	0.9595	0.0175	0.9259
0.36	0.9819	0.36	0.9468	0.018	0.9595	0.018	0.9259
0.37	0.9828	0.37	0.9476	0.0185	0.9595	0.0185	0.9259
0.38	0.9836	0.38	0.9484	0.019	0.9595	0.019	0.9259
0.39	0.9843	0.39	0.9492	0.0195	0.9595	0.0195	0.9259
0.4	0.985	0.4	0.95	0.02	0.9595	0.02	0.9259
0.41	0.9857	0.41	0.951	0.0205	0.9595	0.0205	0.9259
0.42	0.9865	0.42	0.952	0.021	0.9595	0.021	0.9259
0.43	0.9871	0.43	0.9531	0.0215	0.9595	0.0215	0.9259
0.44	0.9877	0.44	0.9542	0.022	0.9595	0.022	0.9259
0.45	0.9882	0.45	0.9552	0.0225	0.9694	0.0225	0.9358
0.46	0.9887	0.46	0.9562	0.023	0.97	0.023	0.9364
0.47	0.9891	0.47	0.9572	0.0235	0.9795	0.0235	0.9459
0.48	0.9894	0.48	0.9581	0.024	0.9795	0.024	0.9459
0.49	0.9897	0.49	0.959	0.0245	0.9795	0.0245	0.9459
0.5	0.99	0.5	0.96	0.025	0.9795	0.025	0.9459
0.51	0.9904	0.51	0.961	0.0255	0.9795	0.0255	0.9459
0.52	0.9908	0.52	0.9621	0.026	0.9795	0.026	0.9459
0.53	0.9911	0.53	0.9631	0.0265	0.989	0.0265	0.9554
0.54	0.9915	0.54	0.9641	0.027	0.9891	0.027	0.9558
0.55	0.9918	0.55	0.9651	0.0275	0.9891	0.0275	0.9558
0.56	0.9921	0.56	0.9661	0.028	0.9894	0.028	0.9561
0.57	0.9923	0.57	0.9671	0.0285	0.9894	0.0285	0.9561
0.58	0.9926	0.58	0.9681	0.029	0.9894	0.029	0.9561
0.59	0.9928	0.59	0.969	0.0295	0.9894	0.0295	0.9561
0.6	0.993	0.6	0.97	0.03	0.9894	0.03	0.9561
0.61	0.9932	0.61	0.971	0.0305	0.9894	0.0305	0.9561
0.62	0.9935	0.62	0.972	0.031	0.9894	0.031	0.9561
0.63	0.9937	0.63	0.973	0.0315	0.9894	0.0315	0.9561
0.64	0.994	0.64	0.974	0.032	0.9894	0.032	0.9561
0.65	0.9942	0.65	0.975	0.0325	0.9894	0.0325	0.9561
0.66	0.9944	0.66	0.976	0.033	0.9898	0.033	0.9565
0.67	0.9945	0.67	0.977	0.0335	0.9898	0.0335	0.9565
0.68	0.9947	0.68	0.978	0.034	0.9898	0.034	0.9565
0.69	0.9948	0.69	0.979	0.0345	0.9898	0.0345	0.9565

0.7	0.995	0.7	0.98	0.035	0.9898	0.035	0.9565
0.71	0.9952	0.71	0.9811	0.0355	0.9902	0.0355	0.9569
0.72	0.9953	0.72	0.9821	0.036	0.9902	0.036	0.9569
0.73	0.9955	0.73	0.9832	0.0365	0.9902	0.0365	0.9569
0.74	0.9956	0.74	0.9842	0.037	0.9902	0.037	0.9569
0.75	0.9958	0.75	0.9853	0.0375	0.9902	0.0375	0.9569
0.76	0.9959	0.76	0.9863	0.038	0.9906	0.038	0.9573
0.77	0.9961	0.77	0.9873	0.0385	0.9906	0.0385	0.9573
0.78	0.9962	0.78	0.9883	0.039	0.9906	0.039	0.9573
0.79	0.9964	0.79	0.9892	0.0395	0.9994	0.0395	0.9661
0.8	0.9965	0.8	0.99	0.04	0.9994	0.04	0.9661
0.81	0.9966	0.81	0.9907	0.0405	0.9997	0.0405	0.9664
0.82	0.9967	0.82	0.9914	0.041	0.9997	0.041	0.9664
0.83	0.9969	0.83	0.992	0.0415	0.9997	0.0415	0.9664
0.84	0.997	0.84	0.9926	0.042	0.9997	0.042	0.9664
0.85	0.9971	0.85	0.9931	0.0425	0.9997	0.0425	0.9664
0.86	0.9972	0.86	0.9936	0.043	0.9997	0.043	0.9664
0.87	0.9972	0.87	0.9941	0.0435	0.9997	0.0435	0.9664
0.88	0.9973	0.88	0.9945	0.044	0.9997	0.044	0.9664
0.89	0.9972	0.89	0.9948	0.0445	0.9997	0.0445	0.9664
0.9	0.997	0.9	0.995	0.045	0.9997	0.045	0.9664
0.91	0.9965	0.91	0.9949	0.0455	0.9997	0.0455	0.9664
0.92	0.9958	0.92	0.9948	0.046	1	0.046	0.9667
0.93	0.9952	0.93	0.9946	0.0465	1	0.0465	0.9667
0.94	0.9948	0.94	0.9946	0.047	1	0.047	0.9667
0.95	0.9945	0.95	0.9947	0.0475	1	0.0475	0.9667
0.96	0.9946	0.96	0.995	0.048	1	0.048	0.9667
0.97	0.995	0.97	0.9956	0.0485	1	0.0485	0.9667
0.98	0.996	0.98	0.9965	0.049	1	0.049	0.9667
0.99	0.9976	0.99	0.998	0.0495	1	0.0495	0.9667
1	1	1	1	0.05	1	0.05	0.9667

Bibliography

1. Shannon, R.E. *Systems Simulation: The Art and Science*. Englewood Cliffs, N.J.: Prentice-Hall, 1975.

2. Defense science and technology strategy and plans, "Combat Identification," 2000. [Hhttp://www.wslfweb.org/docs/dstp2000/jwstppdf/08-CID.pdfH; Accessed Sep 4, 2008].

3. Miller, John O. Class Slides, OPER 671, Combat Modeling I. Department of Operations Research, Air Force Institute of Technology, Wright-Patterson AFB OH, July 2008.

4. Kim, Taeho *Combat Identification Modeling Using Robust Optimization Techniques.* MS thesis, AFIT/GOR/ENS/08-11. Department of Operations Research, Air force Institute of Technology (AU), Wright-Patterson AFB OH, 2008

5. George Mason University, "Combat Identification with Bayesian Networks." [http://ite.gmu.edu/~klaskey/papers/LaskeyCCRTSCombatID.pdfH; Accessed Sep 4, 2008].

6. Hartman, James K. Course notes, OPER 671, Combat Modeling I, Lecture Notes in Aggregated Combat Modeling. Air Force Institute of Technology, Wright-Patterson AFB OH, 1985.

7. Fawcett, Tom. "An introduction to ROC analysis," *Pattern Recognition Letters*, 27 (December 2005).

8. Law Averill M. *Simulation modeling and analysis* (4th Edition). New York: McGraw-Hill Book Company, 2007.

9. Robert E. Marks, and Hermann Schnabl, "Genetic Algorithm and Neural Networks: A Comparison Based on the Repeated Prisoner's Dilemma"(May 1998)

10. Chih-Hsien Chen, Hsu-Hwa Chang, and Lee-Ming, "Robust Parameter Design by Neural networks and Genetic Algorithms," *The 24th Workshop on combinational Mathematics and Computation Theory*

11. Montgomery, Douglas C. *Design and Analysis of Experiments* (6th Edition). New Jersey: Wiley, 2005.

12. Nocedal Jorge and Wright, Stephen J. Numerical Optimization. New York: Springer, 2006.

13. Rymond H. Myers and Douglas C. Montgomery. Response Surface Methodology (2nd Edition): Wiley, 2002.

14. Globalsecurity.org [http://www.globalsecurity.org/military/systems/ground/cid.htm: Accessed Feb 2009]

15. Tim Callinan *Artificial Neural Network identification and control of the inverted pendulumTechniques,* 2003

16. DTIC online, “Combat Identification Issues and Challenges,” 2001. [http://www.dtic.mil/ndia/2001ewc/jensen.pdf: Accessed Feb 24, 2009].

17. Jeffery R. Cares, “An Information Age Combat Models” (Sep 2004)

18. Simon Haykin. Neural Network: A comprehensive foundation. New Jersey: Prentice-hall, 1999.

Vita

Captain Changwook Lim graduated from Tae-sung high school in Yongin, Korea. He entered undergraduate studies at the Korea Military Academy, Seoul where he graduated with a Bachelor of Engineering degree in Civil Engineering, and he was commissioned in March 2000.

His first assignment was at 7th Division as an engineer platoon leader in August 2000. In 2004, he did field exercise as a company commander with 50th float-bridge company of U.S Army. In 2005, he participated in Ulji Focus Lense (UFL) which is the biggest combined exercise between U.S. Force and Korea force. In August 2002, he entered the Graduate School of Operations and Researches, Air Force Institute of Technology. Upon graduation, he will be assigned to the 3rd field army in Korea.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
05-03-2009	**Master's Thesis**	June 2008 - Mar 2009

4. TITLE AND SUBTITLE	
COMBAT IDENTIFICATION MODELING USING NEURAL NETWORK TECHNIQUES	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	**5d. PROJECT NUMBER**
Lim, Changwook, Captain, ROKA	**5e. TASK NUMBER**
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Street, Building 642 WPAFB OH 45433-7765	AFIT/GOR/ENS/09-09
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
ACC/A8SI, AFOSR Attn: Charles Sadowski Jr. 204 Dodd Blvd Suite 226 Langley AFB, Va 23665-2702	**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The purposes of this research were: (1) validating Kim's (2007) simulation method by applying analytic methods and (2) comparing the two different Robust Parameter Design methods with three measures of performance (label accuracy for enemy, friendly, and clutter). Considering the features of CID, input variables were defined as two controllable (threshold combination of detector and classifier) and three uncontrollable (map size, number of enemies and friendly). The first set of experiments considers Kim's method using analytical methods. In order to create response variables, Kim's method uses Monte Carlo simulation. The output results showed no difference between simulation and the analytic method. The second set of experiments compared the measures of performance between a standard RPD used by Kim and a new method using Artificial Neural Networks (ANNs). To find optimal combinations of detection and classification thresholds, Kim's model uses regression with a combined array design, whereas the ANNs method uses ANN with a crossed array design. In the case of label accuracy for enemy, Kim's solution showed the higher expected value, however it also showed a higher variance. Additionally, the model's residuals were higher for Kim's model.

15. SUBJECT TERMS
Combat Modeling, Combat Identification, Receiver Operating Characteristics Analysis, Validation, Analytic Model Crossed Array Design, Regression, Neural Network, Label Accuracy

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr. Kenneth W. Bauer
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	**U**	**U**	**UU**	117	(937) 255-6565, ext 4328; e-mail: Kenneth.Bauer@afit.af.mil

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18